

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 16, 2014

P. Ashwood-Smith  
Huawei Technologies  
R. Iyengar  
T. Tsou  
Huawei Technologies USA  
A. Sajassi  
Cisco Technologies  
M. Boucadair  
C. Jacquenet  
France Telecom  
M. Daikoku  
KDDI corporation  
July 15, 2013

NVO3 Operational Requirements  
draft-ashwood-nvo3-operational-requirement-03

Abstract

This document provides framework and requirements for Network Virtualization over Layer 3 (NVO3) Operations, Administration, and Maintenance (OAM). This document for the most part gathers requirements from existing IETF drafts and RFCs which have already extensively studied this subject for different data planes and layering. As a result this draft is high level and broad. We begin to ask which are truly required for NVO3 and expect the list to be narrowed by the working group as subsequent versions of this draft are created.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. OSI Definitions of OAM . . . . .	3
1.2. Requirements Language . . . . .	5
1.3. Relationship with Other OAM Work . . . . .	5
2. Terminology . . . . .	6
3. NVO3 Reference Model . . . . .	6
4. OAM Framework for NVO3 . . . . .	7
4.1. OAM Layering . . . . .	7
4.2. OAM Domains . . . . .	8
5. NVO3 OAM Requirements . . . . .	9
5.1. Discovery . . . . .	9
5.2. Connectivity Fault Management . . . . .	9
5.2.1. Connectivity Fault Detection . . . . .	9
5.2.2. Connectivity Fault Verification . . . . .	9
5.2.3. Connectivity Fault localization . . . . .	10
5.2.4. Connectivity Fault Notification and Alarm Suppression . . . . .	10
5.3. Frame Loss . . . . .	10
5.4. Frame Delay . . . . .	10
5.5. Frame Delay Variation . . . . .	10
5.6. Frame Throughput . . . . .	10
5.7. Frame Discard . . . . .	10
5.8. Availability . . . . .	11
5.9. Data Path Forwarding . . . . .	11
5.10. Scalability . . . . .	11
5.11. Extensibility . . . . .	11
5.12. Security . . . . .	11
5.13. Transport Independence . . . . .	12
5.14. Application Independence . . . . .	12
5.15. Prioritization . . . . .	12
6. Items for Further Discussion . . . . .	12
7. IANA Considerations . . . . .	14

8. Security Considerations . . . . .	14
9. Acknowledgements . . . . .	14
10. References . . . . .	14
10.1. Normative References . . . . .	14
10.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

This document provides framework and requirements for Network virtualization over Layer 3(NVO3) Operation, Administration, and Maintenance (OAM). Given that this OAM subject is far from new and has been under extensive investigation by various IETF working groups (and several other standards bodies) for many years, this document draws from existing work, starting with [RFC6136]. As a result, sections of [RFC6136] have been reused with minor changes with the permission of the authors.

NVO3 OAM requirements are expected to be a subset of IETF/IEEE etc. work done so far; however, we begin with a full set of requirements and expect to prune them through several iterations of this document.

### 1.1. OSI Definitions of OAM

The scope of OAM for any service and/or transport/network infrastructure technologies can be very broad in nature. OSI has defined the following five generic functional areas commonly abbreviated as "FCAPS" [NM-Standards]:

- o Fault Management,
- o Configuration Management,
- o Accounting Management,
- o Performance Management, and
- o Security Management.

This document focuses on the Fault, Performance and to a limited extent the Configuration Management aspects. Other functional aspects of FCAPS and their relevance (or not) to NVO3 are for further study.

Fault Management can typically be viewed in terms of the following categories:

- o Fault Detection;

- o Fault Verification;
- o Fault Isolation;
- o Fault Notification and Alarm Suppression;
- o Fault Recovery.

Fault detection deals with mechanism(s) that can detect both hard failures such as link and device failures, and soft failures, such as software failure, memory corruption, misconfiguration, etc. Fault detection relies upon a set of mechanisms that first allow the observation of an event, then the use of a protocol to dynamically notify a network/system operator (or management system) about the event occurrence, then the use of diagnostic tools to assess the nature and severity of the fault.

After verifying that a fault has occurred along the data path, it is important to be able to isolate the fault to the level of a given device or link. Therefore, a fault isolation mechanism is needed in Fault Management. A fault notification mechanism should be used in conjunction with a fault detection mechanism to notify the devices upstream and downstream to the fault detection point. The fault notification mechanism should also notify NMS systems.

The terms "upstream" and "backward" are used here to denote the direction(s) from which data traffic is flowing. The terms "downstream" and "forward" denote the direction(s) to which data traffic is forwarded.

For example, when there is a client/server relationship between two layered networks (e.g., the NVO3 layer is a client of the outer IP server layer, while the inner IP layer is a client of the NVO3 server layer 2), fault detection at the server layer may result in the following fault notifications:

- o Sending a forward fault notification from the server layer to the client layer network(s) using the fault notification format appropriate to the client layer.
- o Sending a backward fault notification to the server layer, if applicable, in the reverse direction.
- o Sending a backward fault notification to the client layer, if applicable, in the reverse direction.

Finally, fault recovery deals with recovering from the detected failure by switching to an alternate available data path (depending

on the nature of the fault) using alternate devices or links. In fact, the controller can provision another virtual network, thus automatically resolving the reported problem.

The controller may also directly monitor the status of virtual network components such as Network Virtualization Edge elements (NVEs) [NVO3-framework] in order to respond to their failures. In addition to forward and backward fault notifications, the controller may deliver notifications to a higher level orchestration component, e.g., one responsible for Virtual Machine (VM) provisioning and management.

Note, given that the IP network on which NVO3 resides is usually self healing, it is expected that recovery by the NVO3 layer would not normally be required, although there may be a requirement for that layer to log that the problem has been detected and resolved. The special cases of a static IP overlay network, or possibly of a centrally controlled IP overlay network, may, however, require NVO3 involvement in fault recovery.

Performance Management deals with mechanism(s) that allow determining and measuring the performance of the network/services under consideration. Performance Management can be used to verify the compliance to both the service-level and network-level metric objectives/specifications. Performance Management typically consists of measuring performance metrics, e.g., Frame Loss, Frame Delay, Frame Delay Variation (aka Jitter), Frame throughput, Frame discard, etc., across managed entities when the managed entities are in available state. Performance Management is suspended across unavailable managed entities.

## 1.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 1.3. Relationship with Other OAM Work

This document leverages requirements that originate with other OAM work, specifically the following:

- o [RFC6136] provides a template and some of the high level requirements and introductory wording.
- o [IEEE802.1ag] is expected to provide a subset of the requirements for NVO3 both at the Tenant level and also within the L3 Overlay network.

- o [Y.1731] is expected to provide a subset of the requirements for NVO3 at the Tenant level.
- o Section 3.8 of [NVO3-DP-Reqs] lists several requirements specifically concerning ECMP/LAG.

## 2. Terminology

The terminology defined in [NVO3-framework] and [NVO3-DP-Reqs] is used throughout this document. We introduce no new terminology.

## 3. NVO3 Reference Model

Figure 1 below reproduces the generic NVO3 reference model as per [NVO3-framework].

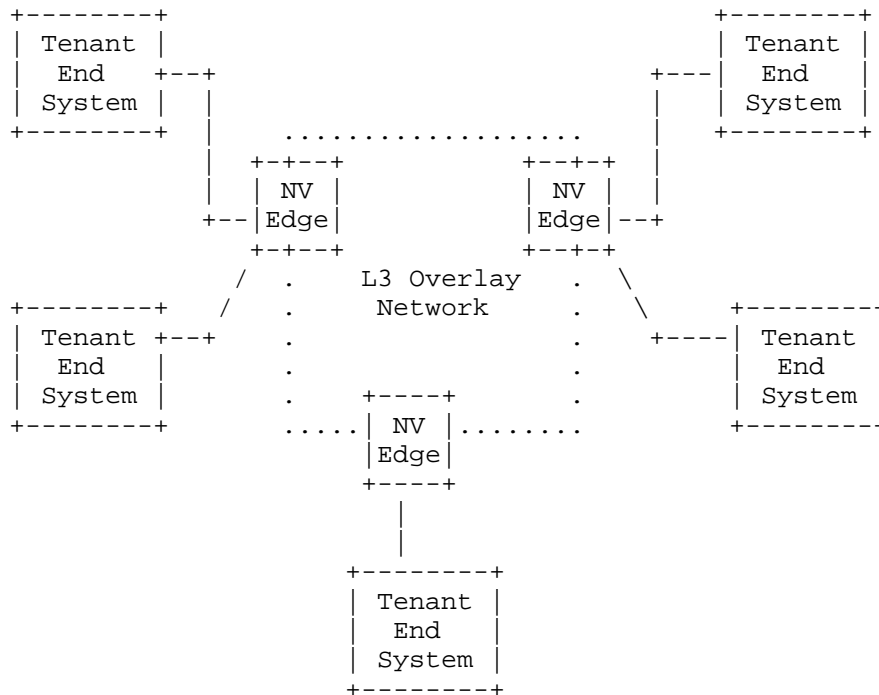


Figure 1: Generic reference model for DC network virtualization over a Layer3 infrastructure

Figure 2 below, reproduces the Generic reference model for the NV Edge (NVE) as per [NVO3-DP-Reqs].

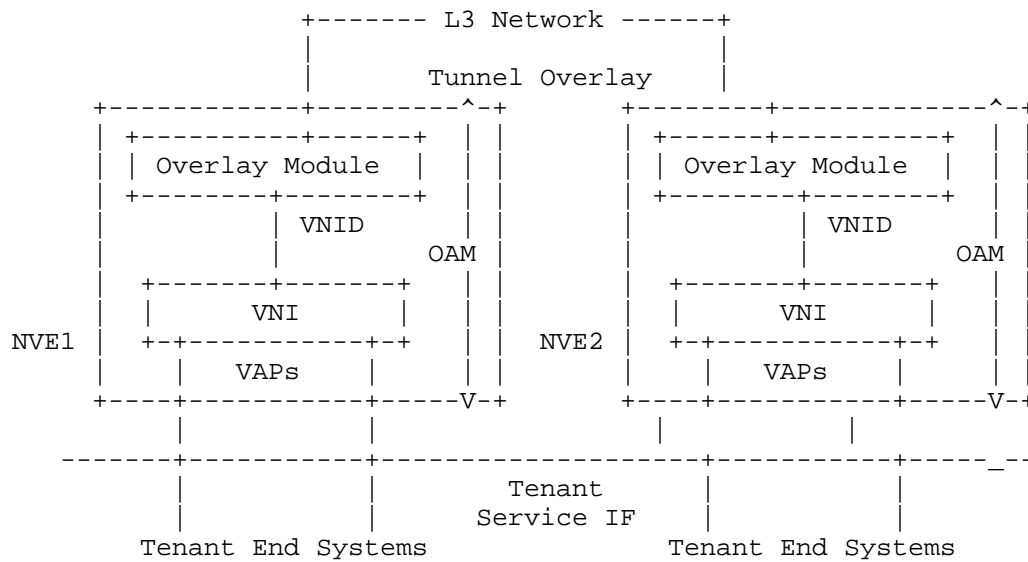


Figure 2: Generic reference model for NV Edge

#### 4. OAM Framework for NVO3

Figure 1 showed the generic reference model for a DC network virtualization over an L3 (or L3VPN) infrastructure while Figure 2 showed the generic reference model for the Network Virtualization (NV) Edge.

L3 network(s) or L3 VPN networks (either IPv6 or IPv4, or a combination thereof), provide transport for an emulated layer 2 created by NV Edge devices. Unicast and multicast tunneling methods (de-multiplexed by Virtual Network Identifier (VNID)) are used to provide connectivity between the NV Edge devices. The NV Edge devices then present an emulated layer 2 network to the Tenant End Systems at a Virtual Network Interface (VNI) through Virtual Access Points (VAPs). The NV Edge devices map layer 2 unicast to layer 3 unicast point-to-point tunnels and may either map layer 2 multicast to layer 3 multicast tunnels or may replicate packets onto multiple layer 3 unicast tunnels.

##### 4.1. OAM Layering

The emulated layer 2 network is provided by the NV Edge devices to which the Tenant End Systems are connected. This network of NV Edges can be operated by a single service provider or can span across multiple administrative domains. Likewise, the L3 Overlay Network can be operated by a single service provider or span across multiple administrative domains.

While each of the layers is responsible for its own OAM, each layer may consist of several different administrative domains. Figure 3 shows an example.

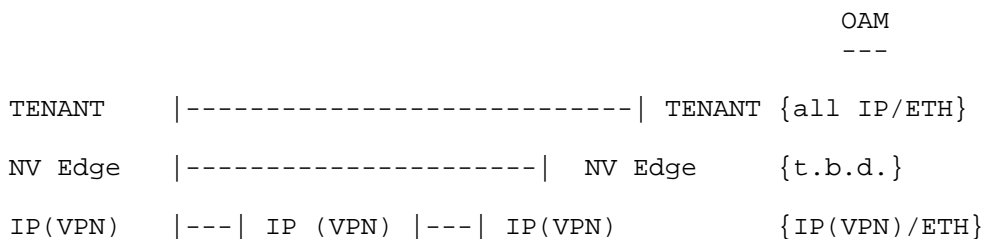


Figure 3: OAM layers in an NVO3 network

For example, at the bottom, at the L3 IP overlay network layer IP(VPN) and/or Ethernet OAM mechanisms are used to probe link by link, node to node etc. OAM addressing here means physical node loopback or interface addresses.

Further up, at the NV Edge layer, NVO3 OAM messages are used to probe the NV Edge to NV Edge tunnels and NV Edge entity status. OAM addressing here likely means the physical node loopback together with the VNI (to de-multiplex the tunnels).

Finally, at the Tenant layer, the IP and/or Ethernet OAM mechanisms are again used but here they are operating over the logical L2/L3 provided by the NV-Edge through the VAP. OAM addressing at this layer deals with the logical interfaces on Vswitches and Virtual Machines.

#### 4.2. OAM Domains

Complex OAM relationships exist as a result of the hierarchical layering of responsibility and of breaking up of end-to-end responsibility.

The OAM domain above NVO3, is expected to be supported by existing IP and L2 OAM methods and tools.



The OAM domain below NVO3, is expected to be supported by existing IP /L2 and MPLS OAM methods and tools. Where this layer is actually multiple domains spliced together, the existing methods to deal with these boundaries are unchanged. Note however that exposing LAG/ECMP detailed behavior may result in additional requirements to this domain, the details of which will be specified in the future versions of this draft.

When we refer to an OAM domain in this document, or just 'domain', we therefore refer to a closed set of NV Edges and the tunnels which interconnect them. Inter-domain OAM considerations will be specified in the future versions of this draft.

## 5. NVO3 OAM Requirements

The following numbered requirements originate from [RFC6136]. All are included however where they seem obviously not relevant (to the present authors) an explanation as to why is included.

### 5.1. Discovery

R1) NVO3 OAM MUST allow an NV Edge device to dynamically discover other NV Edge devices that share the same VNI within a given NVO3 domain. This may be based on a discovery mechanism used to set up data path forwarding between NVEs.

### 5.2. Connectivity Fault Management

#### 5.2.1. Connectivity Fault Detection

R2) NVO3 OAM MUST allow proactive connectivity monitoring between two or more NV Edge devices that support the same VNIs within a given NVO3 domain. NVO3 OAM MAY act as a protection trigger. That is, automatic recovery from transmission facility failure by switchover to a redundant replacement facility may be triggered by notifications from NVO3 OAM.

R3) NVO3 OAM MUST allow monitoring/tracing of all possible paths in the underlay network between a specified set of two or more NV Edge devices. Using this feature, equal cost paths that traverse LAG and/or ECMP may be differentiated.

#### 5.2.2. Connectivity Fault Verification

R4) NVO3 OAM MUST allow connectivity fault verification between two or more NV Edge devices that support the same VNI within a given NVO3 domain.

### 5.2.3. Connectivity Fault localization

R5) NVO3 OAM MUST allow connectivity fault localization between two or more NV Edge devices that support the same VNI within a given NVO3 domain.

### 5.2.4. Connectivity Fault Notification and Alarm Suppression

R6) NVO3 OAM MUST support fault notification to be triggered as a result of the faults occurring in the underneath network infrastructure. This fault notification SHOULD be used for the suppression of redundant service-level alarms.

### 5.3. Frame Loss

R7) NVO3 OAM MUST support measurement of per VNI frame loss between two NV Edge devices that support the same VNI within a given NVO3 domain.

### 5.4. Frame Delay

R8) NVO3 OAM MUST support measurement of per VNI two-way frame delay between two NV edge devices that support the same VNI within a given NVO3 domain.

R9) NVO3 OAM MUST support measurement of per VNI one-way frame delay between two NV Edge devices that support the same VNI within a given NVO3 domain.

### 5.5. Frame Delay Variation

R10) NVO3 OAM MUST support measurement of per VNI frame delay variation between two NV Edge devices that support the same VNI within a given NVO3 domain.

### 5.6. Frame Throughput

R11) NVO3 OAM MAY [\*\*\* Should this be stronger? \*\*\*] support measurement of per VNI frame throughput (in frames and bytes) between two NV Edge devices that support the same VNI within a given NVO3 domain. This feature could be an effective way to confirm whether or not assigned path bandwidth conforms to service level agreement before providing the path between two NV Edge devices.

### 5.7. Frame Discard

R12) NVO3 OAM MAY support measurement of per VNI frame discard between two NV Edge devices that support the same VNI within a given

NVO3 domain. This feature MAY be effective to monitor bursty traffic between two NV Edge devices.

#### 5.8. Availability

A service may be considered unavailable if the service frames/packets do not reach their intended destination (e.g., connectivity is down) or the service is degraded (e.g., frame loss and/or frame delay and/or delay variation threshold is exceeded). Entry and exit conditions may be defined for the unavailable state. Availability itself may be defined in the context of a service type. Since availability measurement may be associated with connectivity, frame loss, frame delay, and frame delay variation measurements, no additional requirements are specified currently.

#### 5.9. Data Path Forwarding

R13) NVO3 OAM frames MUST be forwarded along the same path (i.e., links (including LAG members) and nodes) as the NVO3 data frames.

R14) NVO3 OAM frames MUST provide a mechanism to exercise/trace all data paths that result due to ECMP/LAG hops in the underlay network.

#### 5.10. Scalability

R15) NVO3 OAM MUST be scalable such that an NV edge device can support proactive OAM for each VNI that is supported by the device. (Note - Likely very hard to achieve with hash based ECMP/LAG).

#### 5.11. Extensibility

R16) NVO3 OAM should be extensible such that new functionality and information elements related to this functionality can be introduced in the future.

R17) NVO3 OAM MUST be defined such that devices not supporting the OAM are able to forward the OAM frames in a similar fashion as the regular NVO3 data frames/packets.

#### 5.12. Security

R18) NVO3 OAM frames MUST be prevented from leaking outside their NVO3 domain.

R19) NVO3 OAM frames from outside an NVO3 domain MUST be prevented from entering the said NVO3 domain when such OAM frames belong to the same level or to a lower-level OAM. (Trivially met because hierarchical domains are independent technologies.)

R20) NVO3 OAM frames from outside an NVO3 domain MUST be transported transparently inside the NVO3 domain when such OAM frames belong to a higher-level NVO3 domain. (Trivially met because hierarchical domains are independent technologies).

#### 5.13. Transport Independence

Similar to transport requirement from [RFC6136], we expect NVO3 OAM will leverage the OAM capabilities of the transport layer (e.g., IP underlay).

R21) NVO3 OAM MAY allow adaptation/interworking with its IP underlay OAM functions. For example, this would be useful to allow fault notifications from the IP layer to be sent to the NVO3 layer and likewise exposure of LAG / ECMP will require such non-independence.

#### 5.14. Application Independence

R22) NVO3 OAM MUST [\*\*\* discuss -- is this too strong? \*\*\*] be independent of the application technologies and specific application OAM capabilities.

[Comment -- ECM: Noticed Nicira implementation has a dedicated NVP manager node to play the role of FCAPS here. It is both application layer and OAM layer. May not meet this requirement. In reality, due to the nature of overlay network, very often, vendors are going to make everything all together to a dedicated manager node.]

#### 5.15. Prioritization

R23) NVO3 OAM messages MUST be preferentially treated in NVE and between NVEs, since NVO3 OAM MAY be used to trigger protection switching. As noted above (R2), protection switching is the automatic replacement of a failed transmission facility with a working one providing equal or greater capacity, typically within a few tens of milliseconds from fault detection.

[Comment -- ECM: giving NVO3 OAM messages priority treatment may interfere with measurements of frame delay and jitter.]

### 6. Items for Further Discussion

This section identifies a set of operational items which may be elaborated further if these items fall within the scope of the NVO3.

- o VNID renumbering support

- \* Means to change the VNID assigned to a given instance MUST [\*\*\* discuss: is this too strong? \*\*\*] be supported.
- \* System convergence subsequent to VNID renumbering MUST NOT take longer than a few seconds, to minimize impact on the tenant systems.
- \* A VNE MUST be able to map a VNID with a virtual network context.
- o VNI migration and management operations
  - \* Means to delete an existing VNI MUST be supported.
  - \* Means to add a new VNI MUST be supported.
  - \* Means to merge several VNIs MAY be supported.
  - \* Means to retrieve reporting data per VNI MUST be supported.
  - \* Means to monitor the network resources per VNI MUST be supported.
- o Support of planned maintenance operations on the NVO3 infrastructure
  - \* Graceful procedure to allow for planned maintenance operation on NVE MUST be supported. This includes undoing any configuration changes made for maintenance purposes after completion of the maintenance.
- o Support for communication among virtual networks
  - \* For global reachability purposes, communication among virtual networks MUST be supported. This can be enforced using a NAT function.
- o Activation of new network-related services to the NVO3
  - \* Means to assist in activating new network services (e.g., multicast) without impacting running service should be supported.
- o Inter-operator NVO3 considerations
  - \* As NVO3 may be deployed over inter-operator infrastructure, coordinating OAM actions in each individual domain are required to ensure an end-to-end OAM. In particular, this assumes

existence of agreements on the measurement and monitoring methods, fault detection and repair actions, extending QoS classes (e.g., DSCP mapping policies), etc.

[[DISCUSSION NOTE: Should inter-operator issues be declared out of scope?]]

## 7. IANA Considerations

This memo includes no request to IANA.

## 8. Security Considerations

TBD

## 9. Acknowledgements

The authors are grateful for the contributions of David Black, Dennis Qin, Erik Smith and Ziyi Yang to this latest version.

## 10. References

### 10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

### 10.2. Informative References

[IEEE802.1ag]  
IEEE, "IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks, Amendment 5: Connectivity Fault Management", 2007.

[IEEE802.1ah]  
IEEE, "IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks, Amendment 6: Provider Backbone Bridges", 2008.

[NM-Standards]  
ITU-T, "ITU-T Recommendation M.3400 (02/2000) - TMN Management Functions", February 2000.

[NVO3-DP-Reqs]  
Bitar, N., Lasserre, M., Balus, F., Morin, T., Jin, L., and B. Khasnabish, "NVO3 Data Plane Requirements", October 2012.

## [NVO3-framework]

Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y.  
Rekhter, "Framework for DC Network Virtualization", July  
2012.

[RFC6136] Sajassi, A. and D. Mohan, "Layer 2 Virtual Private Network  
(L2VPN) Operations, Administration, and Maintenance (OAM)  
Requirements and Framework", RFC 6136, March 2011.

[Y.1731] ITU-T, "ITU-T Recommendation Y.1731 (02/08) - OAM  
functions and mechanisms for Ethernet based networks",  
February 2008.

## Authors' Addresses

Peter Ashwood-Smith  
Huawei Technologies  
303 Terry Fox Drive, Suite 400  
Kanata, Ontario K2K 3J1  
Canada

Phone: +1 613 595-1900  
Email: Peter.AshwoodSmith@huawei.com

Ranga Iyengar  
Huawei Technologies USA  
2330 Central Expy  
Santa Clara, CA 95050  
USA

Email: ranga.Iyengar@huawei.com

Tina Tsou  
Huawei Technologies USA  
2330 Central Expy  
Santa Clara, CA 95050  
USA

Email: Tina.Tsou.Zouting@huawei.com

Ali Sajassi  
Cisco Technologies  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Email: [sajassi@cisco.com](mailto:sajassi@cisco.com)

Mohamed Boucadair  
France Telecom  
Rennes 35000  
France

Email: [mohamed.boucadair@orange.com](mailto:mohamed.boucadair@orange.com)

Christian Jacquenet  
France Telecom  
Rennes 35000  
France

Email: [christian.jacquenet@orange.com](mailto:christian.jacquenet@orange.com)

Masahiro Daikoku  
KDDI corporation  
3-10-10, Iidabashi, Chiyoda-ku  
Tokyo 1028460  
Japan

Email: [ms-daikoku@kddi.com](mailto:ms-daikoku@kddi.com)



Internet Engineering Task Force  
Internet Draft  
Intended status: Informational  
Expires: May 2013

Nabil Bitar  
Verizon

Marc Lasserre  
Florin Balus  
Alcatel-Lucent

Thomas Morin  
France Telecom Orange

Lizhong Jin  
Bhumip Khasnabish  
ZTE

November 28, 2012

NVO3 Data Plane Requirements  
draft-bl-nvo3-dataplane-requirements-03.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on May 28, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## Abstract

Several IETF drafts relate to the use of overlay networks to support large scale virtual data centers. This draft provides a list of data plane requirements for Network Virtualization over L3 (NVO3) that have to be addressed in solutions documents.

## Table of Contents

1. Introduction.....	3
1.1. Conventions used in this document.....	3
1.2. General terminology.....	3
2. Data Path Overview.....	4
3. Data Plane Requirements.....	5
3.1. Virtual Access Points (VAPs).....	5
3.2. Virtual Network Instance (VNI).....	5
3.2.1. L2 VNI.....	5
3.2.2. L3 VNI.....	6
3.3. Overlay Module.....	7
3.3.1. NVO3 overlay header.....	8
3.3.1.1. Virtual Network Context Identification.....	8
3.3.1.2. Service QoS identifier.....	8
3.3.2. Tunneling function.....	9
3.3.2.1. LAG and ECMP.....	10
3.3.2.2. DiffServ and ECN marking.....	10
3.3.2.3. Handling of BUM traffic.....	11
3.4. External NVO3 connectivity.....	11
3.4.1. GW Types.....	12
3.4.1.1. VPN and Internet GWs.....	12
3.4.1.2. Inter-DC GW.....	12
3.4.1.3. Intra-DC gateways.....	12

3.4.2. Path optimality between NVEs and Gateways.....	12
3.4.2.1. Triangular Routing Issues,a.k.a.: Traffic Tromboning	13
3.5. Path MTU.....	14
3.6. Hierarchical NVE.....	15
3.7. NVE Multi-Homing Requirements.....	15
3.8. OAM.....	16
3.9. Other considerations.....	16
3.9.1. Data Plane Optimizations.....	16
3.9.2. NVE location trade-offs.....	17
4. Security Considerations.....	17
5. IANA Considerations.....	17
6. References.....	18
6.1. Normative References.....	18
6.2. Informative References.....	18
7. Acknowledgments.....	19

## 1. Introduction

### 1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

### 1.2. General terminology

The terminology defined in [NVO3-framework] is used throughout this document. Terminology specific to this memo is defined here and is introduced as needed in later sections.

DC: Data Center

BUM: Broadcast, Unknown Unicast, Multicast traffic

TS: Tenant System

VAP: Virtual Access Point

VNI: Virtual Network Instance

VNID: VNI ID

## 2. Data Path Overview

The NVO3 framework [NVO3-framework] defines the generic NVE model depicted in Figure 1:

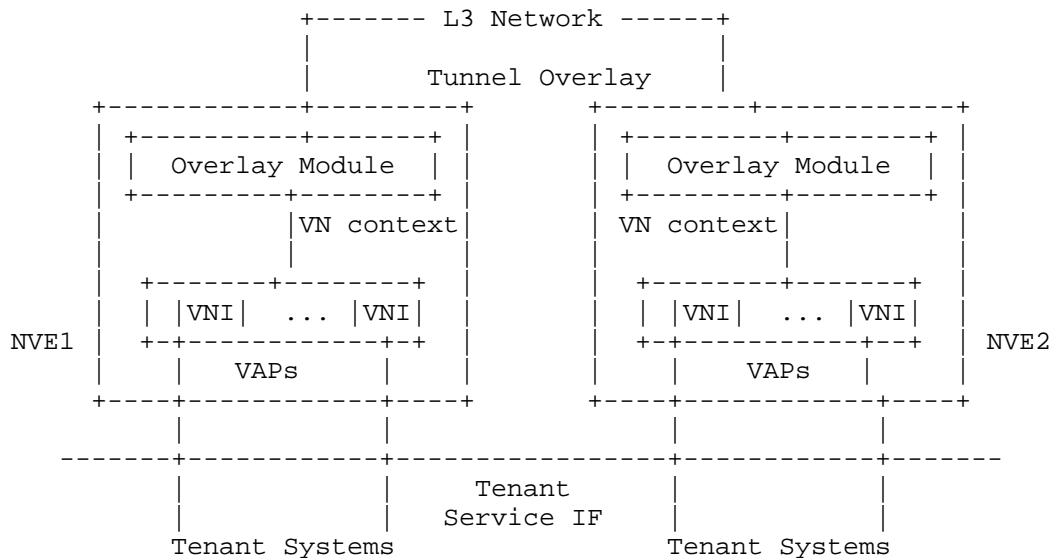


Figure 1 : Generic reference model for NV Edge

When a frame is received by an ingress NVE from a Tenant System over a local VAP, it needs to be parsed in order to identify which virtual network instance it belongs to. The parsing function can examine various fields in the data frame (e.g., VLANID) and/or associated interface/port the frame came from.

Once a corresponding VNI is identified, a lookup is performed to determine where the frame needs to be sent. This lookup can be based on any combinations of various fields in the data frame (e.g., destination MAC addresses and/or destination IP addresses). Note that additional criteria such as 802.1p and/or DSCP markings might be used to select an appropriate tunnel or local VAP destination.

Lookup tables can be populated using different techniques: data plane learning, management plane configuration, or a distributed control plane. Management and control planes are not in the scope of

this document. The data plane based solution is described in this document as it has implications on the data plane processing function.

The result of this lookup yields the corresponding information needed to build the overlay header, as described in section 3.3. This information includes the destination L3 address of the egress NVE. Note that this lookup might yield a list of tunnels such as when ingress replication is used for BUM traffic.

The overlay header MUST include a context identifier which the egress NVE will use to identify which VNI this frame belongs to.

The egress NVE checks the context identifier and removes the encapsulation header and then forwards the original frame towards the appropriate recipient, usually a local VAP.

### 3. Data Plane Requirements

#### 3.1. Virtual Access Points (VAPs)

The NVE forwarding plane MUST support VAP identification through the following mechanisms:

- Using the local interface on which the frames are received, where the local interface may be an internal, virtual port in a VSwitch or a physical port on the ToR
- Using the local interface and some fields in the frame header, e.g. one or multiple VLANs or the source MAC

#### 3.2. Virtual Network Instance (VNI)

VAPs are associated with a specific VNI at service instantiation time.

A VNI identifies a per-tenant private context, i.e. per-tenant policies and a FIB table to allow overlapping address space between tenants.

There are different VNI types differentiated by the virtual network service they provide to Tenant Systems. Network virtualization can be provided by L2 and/or L3 VNIs.

##### 3.2.1. L2 VNI

An L2 VNI MUST provide an emulated Ethernet multipoint service as if Tenant Systems are interconnected by a bridge (but instead by using

a set of NVO3 tunnels). The emulated bridge MAY be 802.1Q enabled (allowing use of VLAN tags as a VAP). An L2 VNI provides per tenant virtual switching instance with MAC addressing isolation and L3 tunneling. Loop avoidance capability MUST be provided.

Forwarding table entries provide mapping information between MAC addresses and L3 tunnel destination addresses. Such entries MAY be populated by a control or management plane, or via data plane.

In the absence of a management or control plane, data plane learning MUST be used to populate forwarding tables. As frames arrive from VAPs or from overlay tunnels, standard MAC learning procedures are used: The source MAC address is learned against the VAP or the NVO3 tunnel on which the frame arrived. This implies that unknown unicast traffic be flooded i.e. broadcast.

When flooding is required, either to deliver unknown unicast, or broadcast or multicast traffic, the NVE MUST either support ingress replication or multicast. In this latter case, the NVE MUST be able to build at least a default flooding tree per VNI. In such cases, multiple VNIs MAY share the same default flooding tree. The flooding tree is equivalent with a multicast (\*,G) construct where all the NVEs for which the corresponding VNI is instantiated are members. The multicast tree MAY be established automatically via routing and signaling or pre-provisioned.

When tenant multicast is supported, it SHOULD also be possible to select whether the NVE provides optimized multicast trees inside the VNI for individual tenant multicast groups or whether the default VNI flooding tree is used. If the former option is selected the VNI SHOULD be able to snoop IGMP/MLD messages in order to efficiently join/prune Tenant System from multicast trees.

### 3.2.2. L3 VNI

L3 VNIs MUST provide virtualized IP routing and forwarding. L3 VNIs MUST support per-tenant forwarding instance with IP addressing isolation and L3 tunneling for interconnecting instances of the same VNI on NVEs.

In the case of L3 VNI, the inner TTL field MUST be decremented by (at least) 1 as if the NVO3 egress NVE was one (or more) hop(s) away. The TTL field in the outer IP header MUST be set to a value appropriate for delivery of the encapsulated frame to the tunnel exit point. Thus, the default behavior MUST be the TTL pipe model where the overlay network looks like one hop to the sending NVE. Configuration of a "uniform" TTL model where the outer tunnel TTL is

set equal to the inner TTL on ingress NVE and the inner TTL is set to the outer TTL value on egress MAY be supported.

L2 and L3 VNIs can be deployed in isolation or in combination to optimize traffic flows per tenant across the overlay network. For example, an L2 VNI may be configured across a number of NVEs to offer L2 multi-point service connectivity while a L3 VNI can be co-located to offer local routing capabilities and gateway functionality. In addition, integrated routing and bridging per tenant MAY be supported on an NVE. An instantiation of such service may be realized by interconnecting an L2 VNI as access to an L3 VNI on the NVE.

The L3 VNI does not require support for Broadcast and Unknown Unicast traffic. The L3 VNI MAY provide support for customer multicast groups. When multicast is supported, it SHOULD be possible to select whether the NVE provides optimized multicast trees inside the VNI for individual tenant multicast groups or whether a default VNI multicasting tree, where all the NVEs of the corresponding VNI are members, is used.

### 3.3. Overlay Module

The overlay module performs a number of functions related to NVO3 header and tunnel processing.

The following figure shows a generic NVO3 encapsulated frame:

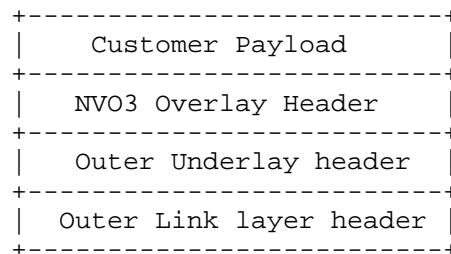


Figure 2 : NVO3 encapsulated frame

where

- . Customer payload: Ethernet or IP based upon the VNI type

- . NVO3 overlay header: Header containing VNI context information and other optional fields that can be used for processing this packet.
- . Outer underlay header: Can be either IP or MPLS
- . Outer link layer header: Header specific to the physical transmission link used

### 3.3.1. NVO3 overlay header

An NVO3 overlay header **MUST** be included after the underlay tunnel header when forwarding tenant traffic. Note that this information can be carried within existing protocol headers (when overloading of specific fields is possible) or within a separate header.

#### 3.3.1.1. Virtual Network Context Identification

The overlay encapsulation header **MUST** contain a field which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE. The egress NVE uses this field to determine the appropriate virtual network context in which to process the packet. This field **MAY** be an explicit, unique (to the administrative domain) virtual network identifier (VNID) or **MAY** express the necessary context information in other ways (e.g. a locally significant identifier).

It **SHOULD** be aligned on a 32-bit boundary so as to make it efficiently processable by the data path. It **MUST** be distributable by a control-plane or configured via a management plane.

In the case of a global identifier, this field **MUST** be large enough to scale to 100's of thousands of virtual networks. Note that there is no such constraint when using a local identifier.

#### 3.3.1.2. Service QoS identifier

Traffic flows originating from different applications could rely on differentiated forwarding treatment to meet end-to-end availability and performance objectives. Such applications may span across one or more overlay networks. To enable such treatment, support for multiple Classes of Service across or between overlay networks **MAY** be required.



To effectively enforce CoS across or between overlay networks, NVEs MAY be able to map CoS markings between networking layers, e.g., Tenant Systems, Overlays, and/or Underlay, enabling each networking layer to independently enforce its own CoS policies. For example:

- TS (e.g. VM) CoS
  - o Tenant CoS policies MAY be defined by Tenant administrators
  - o QoS fields (e.g. IP DSCP and/or Ethernet 802.1p) in the tenant frame are used to indicate application level CoS requirements
- NVE CoS
  - o NVE MAY classify packets based on Tenant CoS markings or other mechanisms (eg. DPI) to identify the proper service CoS to be applied across the overlay network
  - o NVE service CoS levels are normalized to a common set (for example 8 levels) across multiple tenants; NVE uses per tenant policies to map Tenant CoS to the normalized service CoS fields in the NVO3 header
- Underlay CoS
  - o The underlay/core network MAY use a different CoS set (for example 4 levels) than the NVE CoS as the core devices MAY have different QoS capabilities compared with NVEs.
  - o The Underlay CoS MAY also change as the NVO3 tunnels pass between different domains.

Support for NVE Service CoS MAY be provided through a QoS field, inside the NVO3 overlay header. Examples of service CoS provided part of the service tag are 802.1p and DE bits in the VLAN and PBB ISID tags and MPLS TC bits in the VPN labels.

### 3.3.2. Tunneling function

This section describes the underlay tunneling requirements. From an encapsulation perspective, IPv4 or IPv6 MUST be supported, both IPv4 and IPv6 SHOULD be supported, MPLS tunneling MAY be supported.

### 3.3.2.1. LAG and ECMP

For performance reasons, multipath over LAG and ECMP paths SHOULD be supported.

LAG (Link Aggregation Group) [IEEE 802.1AX-2008] and ECMP (Equal Cost Multi Path) are commonly used techniques to perform load-balancing of microflows over a set of a parallel links either at Layer-2 (LAG) or Layer-3 (ECMP). Existing deployed hardware implementations of LAG and ECMP uses a hash of various fields in the encapsulation (outermost) header(s) (e.g. source and destination MAC addresses for non-IP traffic, source and destination IP addresses, L4 protocol, L4 source and destination port numbers, etc). Furthermore, hardware deployed for the underlay network(s) will be most often unaware of the carried, innermost L2 frames or L3 packets transmitted by the TS. Thus, in order to perform fine-grained load-balancing over LAG and ECMP paths in the underlying network, the encapsulation MUST result in sufficient entropy to exercise all paths through several LAG/ECMP hops. The entropy information MAY be inferred from the NVO3 overlay header or underlay header.

All packets that belong to a specific flow MUST follow the same path in order to prevent packet re-ordering. This is typically achieved by ensuring that the fields used for hashing are identical for a given flow.

All paths available to the overlay network SHOULD be used efficiently. Different flows SHOULD be distributed as evenly as possible across multiple underlay network paths. For instance, this can be achieved by ensuring that some fields used for hashing are randomly generated.

### 3.3.2.2. DiffServ and ECN marking

When traffic is encapsulated in a tunnel header, there are numerous options as to how the Diffserv Code-Point (DSCP) and Explicit Congestion Notification (ECN) markings are set in the outer header and propagated to the inner header on decapsulation.

[RFC2983] defines two modes for mapping the DSCP markings from inner to outer headers and vice versa. The Uniform model copies the inner DSCP marking to the outer header on tunnel ingress, and copies that outer header value back to the inner header at tunnel egress. The Pipe model sets the DSCP value to some value based on local policy at ingress and does not modify the inner header on egress. Both models SHOULD be supported.

ECN marking MUST be performed according to [RFC6040] which describes the correct ECN behavior for IP tunnels.

### 3.3.2.3. Handling of BUM traffic

NVO3 data plane support for either ingress replication or point-to-multipoint tunnels is required to send traffic destined to multiple locations on a per-VNI basis (e.g. L2/L3 multicast traffic, L2 broadcast and unknown unicast traffic). It is possible that both methods be used simultaneously.

There is a bandwidth vs state trade-off between the two approaches. User-definable knobs MUST be provided to select which method(s) gets used based upon the amount of replication required (i.e. the number of hosts per group), the amount of multicast state to maintain, the duration of multicast flows and the scalability of multicast protocols.

When ingress replication is used, NVEs MUST track for each VNI the related tunnel endpoints to which it needs to replicate the frame.

For point-to-multipoint tunnels, the bandwidth efficiency is increased at the cost of more state in the Core nodes. The ability to auto-discover or pre-provision the mapping between VNI multicast trees to related tunnel endpoints at the NVE and/or throughout the core SHOULD be supported.

### 3.4. External NVO3 connectivity

NVO3 services MUST interoperate with current VPN and Internet services. This may happen inside one DC during a migration phase or as NVO3 services are delivered to the outside world via Internet or VPN gateways.

Moreover the compute and storage services delivered by a NVO3 domain may span multiple DCs requiring Inter-DC connectivity. From a DC perspective a set of gateway devices are required in all of these cases albeit with different functionalities influenced by the overlay type across the WAN, the service type and the DC network technologies used at each DC site.

A GW handling the connectivity between NVO3 and external domains represents a single point of failure that may affect multiple tenant services. Redundancy between NVO3 and external domains MUST be supported.

### 3.4.1. GW Types

#### 3.4.1.1. VPN and Internet GWs

Tenant sites may be already interconnected using one of the existing VPN services and technologies (VPLS or IP VPN). If a new NVO3 encapsulation is used, a VPN GW is required to forward traffic between NVO3 and VPN domains. Translation of encapsulations MAY be required. Internet connected Tenants require translation from NVO3 encapsulation to IP in the NVO3 gateway. The translation function SHOULD NOT require provisioning touches and SHOULD NOT use intermediate hand-offs, for example VLANs.

#### 3.4.1.2. Inter-DC GW

Inter-DC connectivity MAY be required to provide support for features like disaster prevention or compute load re-distribution. This MAY be provided via a set of gateways interconnected through a WAN. This type of connectivity MAY be provided either through extension of the NVO3 tunneling domain or via VPN GWs.

#### 3.4.1.3. Intra-DC gateways

Even within one DC there may be End Devices that do not support NVO3 encapsulation, for example bare metal servers, hardware appliances and storage. A gateway device, e.g. a ToR, is required to translate the NVO3 to Ethernet VLAN encapsulation.

### 3.4.2. Path optimality between NVEs and Gateways

Within the NVO3 overlay, a default assumption is that NVO3 traffic will be equally load-balanced across the underlying network consisting of LAG and/or ECMP paths. This assumption is valid only as long as: a) all traffic is load-balanced equally among each of the component-links and paths; and, b) each of the component-links/paths is of identical capacity. During the course of normal operation of the underlying network, it is possible that one, or more, of the component-links/paths of a LAG may be taken out-of-service in order to be repaired, e.g.: due to hardware failure of cabling, optics, etc. In such cases, the administrator should configure the underlying network such that an entire LAG bundle in the underlying network will be reported as operationally down if there is a failure of any single component-link member of the LAG bundle, (e.g.: N = M configuration of the LAG bundle), and, thus, they know that traffic will be carried sufficiently by alternate, available (potentially ECMP) paths in the underlying network. This is a likely an adequate assumption for Intra-DC traffic where

presumably the costs for additional, protection capacity along alternate paths is not cost-prohibitive. Thus, there are likely no additional requirements on NVO3 solutions to accommodate this type of underlying network configuration and administration.

There is a similar case with ECMP, used Intra-DC, where failure of a single component-path of an ECMP group would result in traffic shifting onto the surviving members of the ECMP group. Unfortunately, there are no automatic recovery methods in IP routing protocols to detect a simultaneous failure of more than one component-path in a ECMP group, operationally disable the entire ECMP group and allow traffic to shift onto alternative paths. This problem is attributable to the underlying network and, thus, out-of-scope of any NVO3 solutions.

On the other hand, for Inter-DC and DC to External Network cases that use a WAN, the costs of the underlying network and/or service (e.g.: IPVPN service) are more expensive; therefore, there is a requirement on administrators to both: a) ensure high availability (active-backup failover or active-active load-balancing); and, b) maintaining substantial utilization of the WAN transport capacity at nearly all times, particularly in the case of active-active load-balancing. With respect to the dataplane requirements of NVO3 solutions, in the case of active-backup fail-over, all of the ingress NVE's MUST dynamically adapt to the failure of an active NVE GW when the backup NVE GW announces itself into the NVO3 overlay immediately following a failure of the previously active NVE GW and update their forwarding tables accordingly, (e.g.: perhaps through dataplane learning and/or translation of a gratuitous ARP, IPv6 Router Advertisement, etc.) Note that active-backup fail-over could be used to accomplish a crude form of load-balancing by, for example, manually configuring each tenant to use a different NVE GW, in a round-robin fashion. On the other hand, with respect to active-active load-balancing across physically separate NVE GW's (e.g.: two, separate chassis) an NVO3 solution SHOULD support forwarding tables that can simultaneously map a single egress NVE to more than one NVO3 tunnels. The granularity of such mappings, in both active-backup and active-active, MUST be unique to each tenant.

#### 3.4.2.1. Triangular Routing Issues,a.k.a.: Traffic Tromboning

L2/ELAN over NVO3 service may span multiple racks distributed across different DC regions. Multiple ELANs belonging to one tenant may be interconnected or connected to the outside world through multiple Router/VRF gateways distributed throughout the DC regions. In this scenario, without aid from an NVO3 or other type of solution, traffic from an ingress NVE destined to External gateways will take

a non-optimal path that will result in higher latency and costs, (since it is using more expensive resources of a WAN). In the case of traffic from an IP/MPLS network destined toward the entrance to an NVO3 overlay, well-known IP routing techniques MAY be used to optimize traffic into the NVO3 overlay, (at the expense of additional routes in the IP/MPLS network). In summary, these issues are well known as triangular routing.

Procedures for gateway selection to avoid triangular routing issues SHOULD be provided. The details of such procedures are, most likely, part of the NVO3 Management and/or Control Plane requirements and, thus, out of scope of this document. However, a key requirement on the dataplane of any NVO3 solution to avoid triangular routing is stated above, in Section 3.4.2, with respect to active-active load-balancing. More specifically, an NVO3 solution SHOULD support forwarding tables that can simultaneously map a single egress NVE to more than one NVO3 tunnels. The expectation is that, through the Control and/or Management Planes, this mapping information MAY be dynamically manipulated to, for example, provide the closest geographic and/or topological exit point (egress NVE) for each ingress NVE.

### 3.5. Path MTU

The tunnel overlay header can cause the MTU of the path to the egress tunnel endpoint to be exceeded.

IP fragmentation SHOULD be avoided for performance reasons.

The interface MTU as seen by a Tenant System SHOULD be adjusted such that no fragmentation is needed. This can be achieved by configuration or be discovered dynamically.

Either of the following options MUST be supported:

- o Classical ICMP-based MTU Path Discovery [RFC1191] [RFC1981] or Extended MTU Path Discovery techniques such as defined in [RFC4821]
- o Segmentation and reassembly support from the overlay layer operations without relying on the Tenant Systems to know about the end-to-end MTU
- o The underlay network MAY be designed in such a way that the MTU can accommodate the extra tunnel overhead.

### 3.6. Hierarchical NVE

It might be desirable to support the concept of hierarchical NVEs, such as spoke NVEs and hub NVEs, in order to address possible NVE performance limitations and service connectivity optimizations.

For instance, spoke NVE functionality MAY be used when processing capabilities are limited. A hub NVE would provide additional data processing capabilities such as packet replication.

NVEs can be either connected in an any-to-any or hub and spoke topology on a per VNI basis.

### 3.7. NVE Multi-Homing Requirements

Multi-homing techniques SHOULD be used to increase the reliability of an nvo3 network. It is also important to ensure that physical diversity in an nvo3 network is taken into account to avoid single points of failure.

Multi-homing can be enabled in various nodes, from tenant systems into TORs, TORs into core switches/routers, and core nodes into DC GWs.

Tenant systems can either be L2 or L3 nodes. In the former case (L2), techniques such as LAG or STP for instance MAY be used. In the latter case (L3), it is possible that no dynamic routing protocol is enabled. Tenant systems can be multi-homed into remote NVE using several interfaces (physical NICS or vNICS) with an IP address per interface either to the same nvo3 network or into different nvo3 networks. When one of the links fails, the corresponding IP is not reachable but the other interfaces can still be used. When a tenant system is co-located with an NVE, IP routing can be relied upon to handle routing over diverse links to TORs.

External connectivity MAY be handled by two or more nvo3 gateways. Each gateway is connected to a different domain (e.g. ISP) and runs BGP multi-homing. They serve as an access point to external networks such as VPNs or the Internet. When a connection to an upstream router is lost, the alternative connection is used and the failed route withdrawn.

### 3.8. OAM

NVE MAY be able to originate/terminate OAM messages for connectivity verification, performance monitoring, statistic gathering and fault isolation. Depending on configuration, NVEs SHOULD be able to process or transparently tunnel OAM messages, as well as supporting alarm propagation capabilities.

Given the critical requirement to load-balance NVO3 encapsulated packets over LAG and ECMP paths, it will be equally critical to ensure existing and/or new OAM tools allow NVE administrators to proactively and/or reactively monitor the health of various component-links that comprise both LAG and ECMP paths carrying NVO3 encapsulated packets. For example, it will be important that such OAM tools allow NVE administrators to reveal the set of underlying network hops (topology) in order that the underlying network administrators can use this information to quickly perform fault isolation and restore the underlying network.

The NVE MUST provide the ability to reveal the set of ECMP and/or LAG paths used by NVO3 encapsulated packets in the underlying network from an ingress NVE to egress NVE. The NVE MUST provide the ability to provide a "ping"-like functionality that can be used to determine the health (liveness) of remote NVE's or their VNI's. The NVE SHOULD provide a "ping"-like functionality to more expeditiously aid in troubleshooting performance problems, i.e.: blackholing or other types of congestion occurring in the underlying network, for NVO3 encapsulated packets carried over LAG and/or ECMP paths.

### 3.9. Other considerations

#### 3.9.1. Data Plane Optimizations

Data plane forwarding and encapsulation choices SHOULD consider the limitation of possible NVE implementations, specifically in software based implementations (e.g. servers running VSwitches)

NVE SHOULD provide efficient processing of traffic. For instance, packet alignment, the use of offsets to minimize header parsing, padding techniques SHOULD be considered when designing NVO3 encapsulation types.

The NV03 encapsulation/decapsulation processing in software-based NVEs SHOULD make use of hardware assist provided by NICs in order to speed up packet processing.



### 3.9.2. NVE location trade-offs

In the case of DC traffic, traffic originated from a VM is native Ethernet traffic. This traffic can be switched by a local VM switch or ToR switch and then by a DC gateway. The NVE function can be embedded within any of these elements.

The NVE function can be supported in various DC network elements such as a VM, VM switch, ToR switch or DC GW.

The following criteria SHOULD be considered when deciding where the NVE processing boundary happens:

- o Processing and memory requirements
  - o Datapath (e.g. lookups, filtering, encapsulation/decapsulation)
  - o Control plane processing (e.g. routing, signaling, OAM)
- o FIB/RIB size
- o Multicast support
  - o Routing protocols
  - o Packet replication capability
- o Fragmentation support
- o QoS transparency
- o Resiliency

## 4. Security Considerations

This requirements document does not raise in itself any specific security issues.

## 5. IANA Considerations

IANA does not need to take any action for this draft.

## 6. References

### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

### 6.2. Informative References

- [NVOPS] Narten, T. et al, "Problem Statement: Overlays for Network Virtualization", draft-narten-nvo3-overlay-problem-statement (work in progress)
- [NVO3-framework] Lasserre, M. et al, "Framework for DC Network Virtualization", draft-lasserre-nvo3-framework (work in progress)
- [OVCPREQ] Kreeger, L. et al, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp (work in progress)
- [FLOYD] Sally Floyd, Allyn Romanow, "Dynamics of TCP Traffic over ATM Networks", IEEE JSAC, V. 13 N. 4, May 1995
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC1191] Mogul, J. "Path MTU Discovery", RFC1191, November 1990
- [RFC1981] McCann, J. et al, "Path MTU Discovery for IPv6", RFC1981, August 1996
- [RFC4821] Mathis, M. et al, "Packetization Layer Path MTU Discovery", RFC4821, March 2007
- [RFC2983] Black, D. "Diffserv and tunnels", RFC2983, October 2000
- [RFC6040] Briscoe, B. "Tunnelling of Explicit Congestion Notification", RFC6040, November 2010
- [RFC6438] Carpenter, B. et al, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC6438, November 2011
- [RFC6391] Bryant, S. et al, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC6391, November 2011

## 7. Acknowledgments

In addition to the authors the following people have contributed to this document:

Shane Amante, Level3

Dimitrios Stiliadis, Rotem Salomonovitch, Alcatel-Lucent

Larry Kreeger, Cisco

This document was prepared using 2-Word-v2.0.template.dot.

## Authors' Addresses

Nabil Bitar  
Verizon  
40 Sylvan Road  
Waltham, MA 02145  
Email: nabil.bitar@verizon.com

Marc Lasserre  
Alcatel-Lucent  
Email: marc.lasserre@alcatel-lucent.com

Florin Balus  
Alcatel-Lucent  
777 E. Middlefield Road  
Mountain View, CA, USA 94043  
Email: florin.balus@alcatel-lucent.com

Thomas Morin  
France Telecom Orange  
Email: thomas.morin@orange.com

Lizhong Jin  
ZTE  
Email : lizhong.jin@zte.com.cn

Bhumip Khasnabish  
ZTE  
Email : Bhumip.khasnabish@zteusa.com

NVO3  
Internet-Draft  
Intended status: Informational  
Expires: January 5, 2013

B. Carpenter  
Univ. of Auckland  
S. Jiang  
Huawei Technologies Co., Ltd  
July 4, 2012

Layer 3 Addressing Considerations for Network Virtualization Overlays  
draft-carpenter-nvo3-addressing-00

Abstract

This document discusses network layer addressing issues for virtual network overlays in large scale data centres hosting many virtual servers for multiple customers.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 5, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Aspects of addressing in virtual overlay networks . . . . .	3
2.1. Address Independence and Isolation . . . . .	3
2.2. Multiple Data Centres . . . . .	4
2.3. Address mapping . . . . .	4
2.4. Address migration . . . . .	5
2.5. DNS . . . . .	5
2.6. Dual Stack Operation . . . . .	6
3. Consequences for IPv4 address management . . . . .	6
4. Consequences for IPv6 address management . . . . .	7
5. Security Considerations . . . . .	7
6. IANA Considerations . . . . .	7
7. Acknowledgements . . . . .	8
8. Change log [RFC Editor: Please remove] . . . . .	8
9. References . . . . .	8
9.1. Normative References . . . . .	8
9.2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

A common technique in large data centres hosting servers and services for many customers is to use virtual layer 3 network overlays (NVO3) to organise, manage and separate the virtual servers used by individual customers. The related problems are discussed in [I-D.narten-nvo3-overlay-problem-statement], and a framework for the main components of a solution is described in [I-D.lasserre-nvo3-framework].

[Note: In this draft we do not yet give detailed references to the various NVO3 drafts, none of which discuss addressing issues in detail.]

When emulating a large number of virtual hosts on whatever physical network topology is used, probably involving multiple LAN segments, virtual LANs at layer2, and both routers and switches, the question of the IP addressing scheme for the virtual hosts is not trivial. The intention of the present document is to describe the resulting consequences for IP address management in such an environment. Firstly the general aspects are discussed, and then the consequences for both IPv4 and IPv6 addressing schemes are described.

## 2. Aspects of addressing in virtual overlay networks

### 2.1. Address Independence and Isolation

In a typical hosting centre, there will be a number of customers, who are quite possibly mutual competitors who happen to use the same hosting centre. It is essential that virtual hosts assigned to one customer cannot communicate directly with, or even be aware of, virtual hosts assigned to any other customer. It is also essential that virtual networks are operationally independent to the maximum possible extent.

Therefore, to simplify operations by clearly separating independent virtual networks (VNs) from one another, and to enhance both real and perceived confidentiality of each network, it is desirable for the addresses used by each virtual layer 3 network to be allocated and managed independently of all the others. A consequence of this is that it becomes reasonably straightforward to configure layer 3 routing such that traffic from one VN can never unintentionally enter another one, because each network has its own well defined range of addresses. Similarly, it is also reasonably straightforward to configure firewalls or filters to detect and block any unwanted traffic between VNs, even if there is a routing misconfiguration.

For these reasons alone, it is necessary for each VN to have its own well-defined layer 3 address space that is managed independently of all other VNs. This requirement is independent of whether the physical hosts that contain the virtual hosts are on one or more physical or bridged LANs or VLANs. In other words, once the layer 3 topology is virtualised, layer 2 address independence and isolation is neither necessary nor sufficient to guarantee layer 3 address independence and isolation.

It should be understood that independent address allocation does not imply unique addresses. In the IPv4 context, as further discussed below, it is very likely that multiple VNs will use the same ambiguous address space, running over the same physical network infrastructure.

## 2.2. Multiple Data Centres

A given customer might have virtual hosts spread across multiple data centres (DCs). Furthermore, those data centres might be owned and operated by competing enterprises. The only safe assumption is that a single address range cannot span multiple DCs, and that a virtual host being relocated to another DC might need to be renumbered. The addressing scheme for virtual hosts must be compatible with such a situation. Most likely this means extending the requirement for address independence and isolation to cover separate parts of a given customer's total set of virtual servers. In other words the usage scenario for any given customer must be able to deal with virtual hosts in multiple independent and mutually isolated layer 3 address spaces, and with the risk of occasional virtual host renumbering.

This complicates the issues of routing configuration and address filtering. If a VN extends over multiple DCs, VN routing across DC borders must be supported for the address ranges concerned, and address filtering must also be applied in a consistent way at each DC hosting part of a given VN.

## 2.3. Address mapping

Several of the NV03 documents state the need for an address mapping scheme. Some aspects of this are discussed in [I-D.kreeger-nvo3-overlay-cp]. It is generally assumed that an NV03 system will be built using tunnels, and the required mapping is between virtual host addresses and tunnel end points. The addressing scheme for virtual hosts needs to be consistent with the mapping system adopted and whatever dynamic update protocol is used for that mapping.

In the case of a VN that covers multiple DCs, the mapping scheme must

also support multiple DCs. The mapping update protocol will need to exchange mapping information between tunnel endpoints at all DCs involved. This information needs to be specified in some detail, and it must be decided whether this protocol needs to be run per VN or per DC, how this protocol decides which DCs it should talk with, etc.

#### 2.4. Address migration

As discussed in [I-D.narten-nvo3-overlay-problem-statement], current virtual host mechanisms assume that a host's IP address is fixed. If a workload is migrated from one physical host to another, the migration mechanisms assume that existing transport layer associations such as TCP sessions stay alive, and the successful migration of a job in progress relies on this.

As workload conditions change in a large data centre, virtual hosts may need to be migrated from one physical host to another, and quite possibly this will mean moving to a different physical LAN. However, the virtual host address itself should remain constant as just mentioned. The addressing scheme adopted needs to be consistent with this requirement. Another way to view this is as an inverted form of renumbering - instead of the address of a given host changing, a given address is reassigned to a different physical host, thereby representing the move of a given virtual host.

When such a move occurs, there will normally be changes in both the layer 2/layer 3 mapping (given by ARP, Neighbour Discovery, etc.) and the virtual host address to tunnel mapping mentioned above. However, an address which is moved in this way should still remain part of the same aggregate for routing purposes. Otherwise, an immediate change to the routing configuration will be needed as well.

As mentioned above, if a virtual host needs to be migrated between DCs, it might be unavoidable for its virtual address to change. In this case an application layer mechanism will be needed to recover from the resulting loss of transport layer sessions.

#### 2.5. DNS

A Domain Name Service, which resolves queries for hostnames into IP addresses, can reduce the direct dependence of customer applications on IP addresses. If a virtual host is always connected using its hostname, the renumbering issue during inter-DC migrations, mentioned in previous section, would be significantly mitigated. However, this would imply a need for rapid DNS updates.



## 2.6. Dual Stack Operation

In the case of a dual stack deployment, where each virtual host has both an IPv4 and an IPv6 address, there will presumably be some sort of interdependency of the two addressing schemes. At least, the virtual subnet topologies would usually be the same for the two addressing schemes, and virtual hosts would need to migrate simultaneously for IPv4 and IPv6 purposes. If this was not the case, scenarios requiring IPv4/IPv6 interworking might arise unexpectedly, which would be inconvenient and inefficient. A particular case would be the migration of a virtual host between two DCs, one of which supports dual stack and the other of which supports only a single stack.

In general, these situations would require a layer 3 IPv4/IPv6 translator within a VN. This solution should be avoided if possible.

## 3. Consequences for IPv4 address management

In IPv4, it must be assumed that in many if not most cases, the virtual hosts will be numbered out of ambiguous private address space [RFC1918]. The only safe assumption for a general model is that any individual VN may use the same address space as any other. This increases the importance of the requirements for address isolation, independence and mapping. In fact, without address mapping (and in some scenarios network address translation) a large scale IPv4 NV03 system could not be made to work.

Since the IPv4 addresses in use will be ambiguous, management tools must be carefully designed so that operators will never need to rely on addresses alone to identify individual servers. For example, when an address is presented to an operator for any reason, it should always be tagged with some sort of VN identifier. The same goes for any place that an address is logged or stored for any other reason. Legacy software and tools that do not do this should be avoided as much as possible.

An interesting aspect of using, say, Net 10 for every VN instance is that the number of virtual hosts can be quite large, up to  $2^{24}$  (in excess of 16 million). This frees the designer from traditional limits on the size of an IPv4 subnet.

An unavoidable consequence of using RFC1918 addresses is that the virtual hosts, if accessed by outside users, will be hidden behind either an application layer proxy or a NAT. In both cases these might be part of a load balancing system.

#### 4. Consequences for IPv6 address management

In IPv6, there is no concept of ambiguous private space. Each VN can have its own global-scope address prefix. This removes the operational problems casued by ambiguous addresses in IPv4.

Even a basic /64 prefix would allow for more virtual host addresses than would ever be possible in IPv4, so again the designer is not restricted by any absolute limit on subnet size. Nevertheless, it is advisable to use a shorter prefix such as /48 or /56 for each VN, so that a VN can span more than one LAN using standard IPv6 routing without difficulty. It is unclear that tunnels and address mapping are needed for IPv6-based VNs, due to the absence of ambiguous addresses.

A choice can be made between a regular IPv6 prefix from the customer's own IPv6 space or from ISP-assigned space, and a Unique Local Address prefix [RFC4193], [I-D.liu-v6ops-ula-usage-analysis]. The latter has the advantage of needing no administrative procedure before assigning it, and it is also routinely blocked by site and ISP border routers, like an RFC1918 prefix. However, apart from that the choice of IPv6 prefix has little external importance and is mainly a matter of convenience.

There is no requirement for IPv6 prefix translation [RFC6296] between the virtual hosts and any outside users. However, the presence of such translation, or of some form of load balancing, cannot be excluded.

#### 5. Security Considerations

Routing configurations and filters in firewalls and routers should be constructed such that, by default, packets from virtual hosts in one VN cannot be forwarded into another VN. Traffic to and from a given VN should only be allowed for the designated users of that VN, and for the VN management and operations tools.

An independent and isolated addressing scheme is not by itself a security solution. While it might avoid the most trivial and straightforward penetration attempts, it is in no way a substitute for a security solution that responds to specific threat models in the NVO3 situation.

#### 6. IANA Considerations

This document requests no action by IANA.

## 7. Acknowledgements

Valuable comments and contributions were made by ... and others.

This document was produced using the xml2rfc tool [RFC2629].

## 8. Change log [RFC Editor: Please remove]

draft-carpenter-nvo3-addressing-00: original version, 2012-07-04.

## 9. References

### 9.1. Normative References

- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

### 9.2. Informative References

- [I-D.kreeger-nvo3-overlay-cp]  
Black, D., Dutt, D., Kreeger, L., Sridhavan, M., and T. Narten, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp-00 (work in progress), January 2012.
- [I-D.lasserre-nvo3-framework]  
Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-lasserre-nvo3-framework-02 (work in progress), June 2012.
- [I-D.liu-v6ops-ula-usage-analysis]  
Liu, B., Jiang, S., and C. Byrne, "Analysis and recommendation for the ULA usage", draft-liu-v6ops-ula-usage-analysis-02 (work in progress), March 2012.
- [I-D.narten-nvo3-overlay-problem-statement]  
Narten, T., Sridhavan, M., Dutt, D., Black, D., and L.

Kreeger, "Problem Statement: Overlays for Network Virtualization",  
draft-narten-nvo3-overlay-problem-statement-02 (work in progress), June 2012.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

#### Authors' Addresses

Brian Carpenter  
Department of Computer Science  
University of Auckland  
PB 92019  
Auckland, 1142  
New Zealand

Email: [brian.e.carpenter@gmail.com](mailto:brian.e.carpenter@gmail.com)

Sheng Jiang  
Huawei Technologies Co., Ltd  
Q14, Huawei Campus  
No.156 Beiqing Road  
Hai-Dian District, Beijing 100095  
P.R. China

Email: [jiangsheng@huawei.com](mailto:jiangsheng@huawei.com)



NVo3  
Internet Draft  
Intended status: Informational  
Expires: December 2012

L. Dunbar  
Huawei  
June 28, 2012

## Issues of Mobility in DC Overlay network

draft-dunbar-nvo3-overlay-mobility-issues-00.txt

### Abstract

This draft describes the issues introduced by VM mobility in Data center overlay network.

### Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 28, 2011.

### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

## Table of Contents

1. Introduction .....	2
2. Terminology .....	3
3. Issues associated with Multicast in Overlay Network.....	3
4. Issues associated with more than 4k Tenant Separation.....	4
4.1. Collision of local VLAN Identifiers when VMs Move.....	7
4.1.1. Local VIDs Managed by External Controller.....	10
4.1.2. Local VIDs Managed by NVE .....	11
4.2. Tenant Virtual Network separation at the physical gateway routers .....	11
5. Summary and Recommendations.....	12
6. Manageability Considerations.....	13
7. Security Considerations.....	13
8. IANA Considerations .....	13
9. Acknowledgments .....	13
10. References .....	13
Authors' Addresses .....	14
Intellectual Property Statement.....	14
Disclaimer of Validity .....	14

## 1. Introduction

Overlay networks, such as VxLAN, NvGRE, etc, have been proposed to scale networks in Data Center with massive number of hosts as the result of server virtualization and business demand.

Overlay network can hide the massive number of VMs' addresses from the switches/routers in the core (i.e. underlay network).

One of the key requirements stated in [NVo3-problem] is the ability of moving VMs across wider range of locations, which could be

multiple server racks, PODs, or locations, without changing VM's IP/MAC addresses. That means the association of VMs to their corresponding NVE is changing as VMs migrate. This dynamic nature of VM mobility in Data Center introduces new challenges and complications to overlay networks. This draft describes some of the issues introduced by VM migration in overlay environment. The purpose of the draft is to ensure those issues will be addressed by future solutions.

## 2. Terminology

CE: VPN Customer Edge Device

DC: Data Center

DA: Destination Address

EOR: End of Row switches in data center.

VNID: Virtual Network Identifier

NVE: Network Virtualization Edge

PE: VPN Provider Edge Device

SA: Source Address

ToR: Top of Rack Switch. It is also known as access switch.

VM: Virtual Machines

VPLS: Virtual Private LAN Service

## 3. Issues associated with Multicast in Overlay Network

Some data centers avoid the use of IP Multicast due, primarily, to the perceptions of configuration/protocol complexity and multicast scaling limits. There are also many data center operators for whom multicast is critical. Among the latter group, multicast is used for Internet Television (IPTV), market data, cluster load balancing, gaming, just to name a few. The use of multicast in overlay environment can impose some issues to network when VMs move, in particular:



The association between multicast members to NVE becomes dynamic as VMs move. At one moment, all members of a multicast group could be attached to one NVE. At another moment, some members of the multicast group could be attached to different NVEs. Among VMs attached to one NVE, some can send, while others can only receive.

In addition, Overlay, which hides the VM addresses, introduces the IGMP snooping issue in the core. With NVE adding outer header to data frames from VMs (i.e. applications), multicast addresses are hidden from the underlay networks, making switches in the underlay network not being able to snoop on the IGMP reports from multicast members.

For unicast data frames, overlay network edge (e.g. TRILL edge) can learn the inner-outer address mapping by observing data frames passing by. Since multicast address is not placed in the inner-header's SA field of data frame, the learning approach for unicast won't work for multicast in overlay.

TRILL solves the multicast inner-outer address learning issues by creating common multicast trees in the TRILL domain. If TRILL's multicast approach is used for DC with VM mobility, the multicast states maintained by switches/routers in the underlay network have to change as VMs move, which means switches in the underlay network have to be aware of VMs mobility and change multicast states accordingly.

Overall, the VM mobility in overlay environment make multicast more complicated for switches/routers in the underlay network and for NVEs.

#### 4. Issues associated with more than 4k Tenant Separation

The [NVo3-framework] has a good figure showing the logical network seen by each tenant. There are L2 domains being connected by L3 infrastructure. Each tenant can have multiple virtual networks, which are identified IEEE802.1Q compliant 12 bits VLAN ID, under its logical routers (Rtr). Any VMs communicating with peers in different subnets, either within DC or outside DC, will have their L2 MAC address destined towards its local Router (Rtr in the figure below).



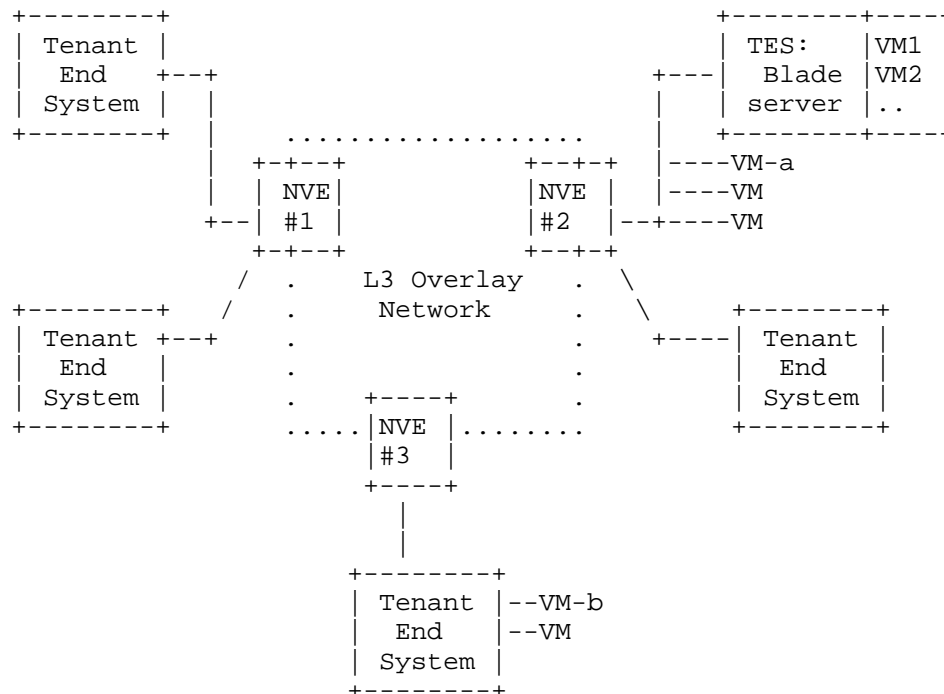


Figure 2: Overlay example

For client traffic "VM-a" to "VM-b", the ingress NVE encapsulates the client payload with an outer header which includes at least egress NVE as DA, ingress NVE as SA, and a VNID. The VNID is a 24-bits identifier proposed by [NVo3-Problem] to separate tens of thousands of tenant virtual networks. When the egress NVE receives the data frame from its ports facing the underlay network, the egress NVE decapsulates the outer header and then forward the decapsulated data frame to the attached VMs.

When "VM-b" is on the same subnet (or VLAN) as "VM-a" and located within the same data center, the corresponding egress NVE is usually on a virtual switch in a server, on a ToR switch, or on a blade switch.

When "VM-b" is on a different subnet (or VLAN), the corresponding egress NVE should be next to (or located on) the logical Rtr (Figure 1), which is most likely located on the data center gateway router(s).

#### 4.1. Collision of local VLAN Identifiers when VMs Move

Since the VMs attached to one NVE could belong to different virtual networks, the traffic under each NVE have to be identified by local network identifiers, which is usually VLAN if VMs are attached to NVE access ports via L2.

To support tens of thousands of virtual networks, the local VID associated with client payload under each NVE has to be locally significant. If ingress NVE simply encapsulates an outer header to data frames received from VMs and forward the encapsulated data frames to egress NVE via underlay network, the egress NVE can't simply decapsulate the outer header and send the decapsulated data frames to attached VMs as done by TRILL. Egress NVE needs to convert the VID carried in the data frame to a local VID for the virtual network before forwarding the data frame to the VMs attached.

In VPLS, operator has to configure the local VIDs under each PE to specific VPN instances. In VPLS, the local VID mapping to VPN instance ID doesn't change very much. In addition, most likely CE is not shared by multiple tenants, so the VIDs on one physical port of PE to CE are only for one tenant. For rare occasion of multiple tenants sharing one CE, the CE can convert the tuple [local customer VIDs & Tenant Access Port] to the VID designated by VPN operator for each VPN instance on the shared link between CE port and PE port. For example, in the figure below, the VIDs under CE#21 and the VIDs under CE#22 can be duplicated as long as the CEs can convert the local VIDs from their downstream links to the VIDs given by the VPN operators for the links between PE and CEs.

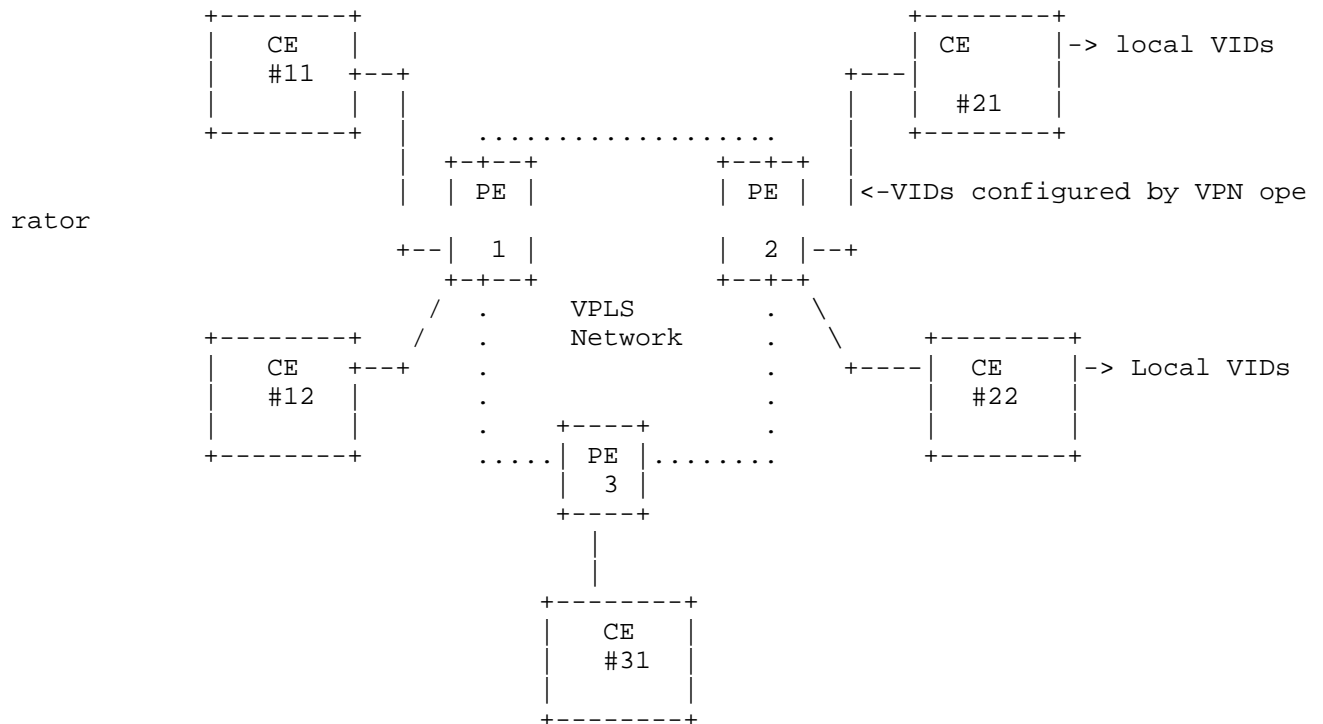


Figure 3: VPLS example

When all VMs under one virtual network are moved away from a NVE, the local VID, which was designated for this virtual network, might need to be used for different virtual network whose VMs are moved in later.

In the Figure below, the NVE#1 may have local VID #100~#200 assigned to some virtual networks attached. The NVE#2 may have local VID #100~#150 assigned to different virtual networks. With VNID encoded in the outer header of data frames, the traffic in the L3 Overlay Network is strictly separated.

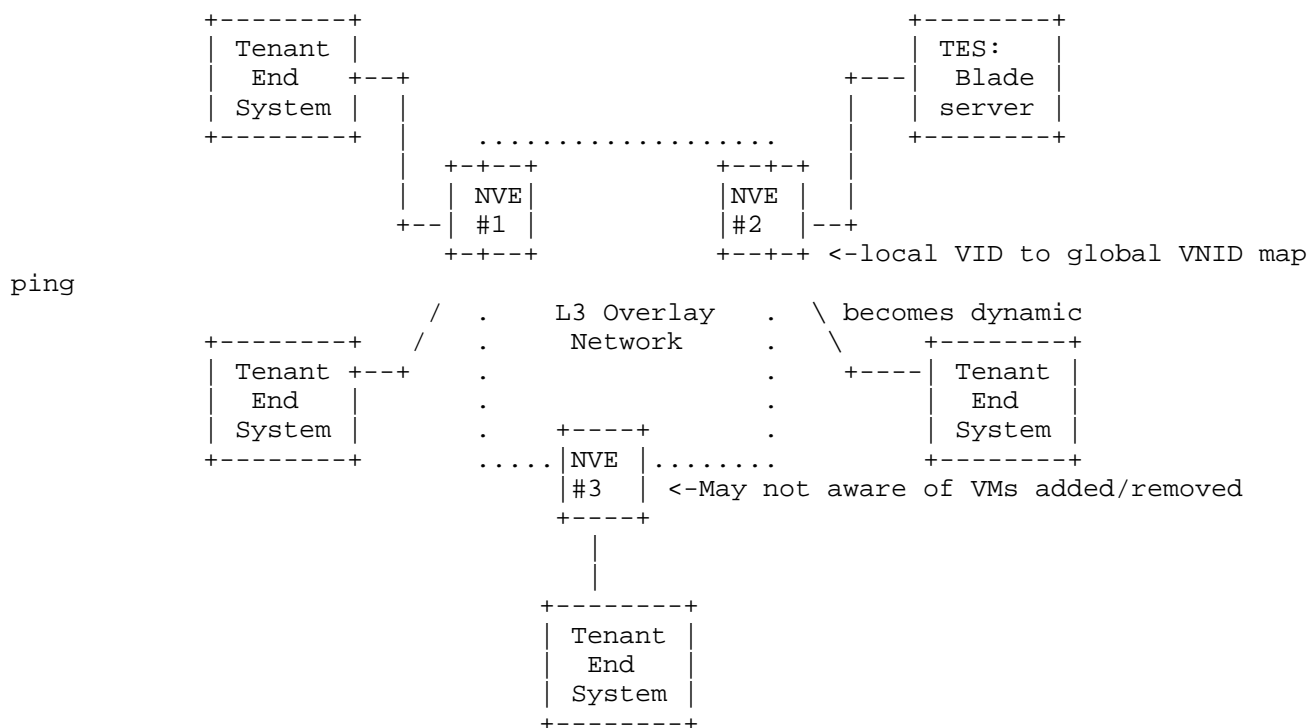


Figure 4: Overlay example

When some VMs associated with Virtual Network X using VID 120 under NVE1 are moved to NVE2, a new VID must be assigned for the Virtual Network X under NVE2.

It gets complicated when the local VIDs are tagged by none-NVE devices, e.g. VMs themselves, blade server switches, or virtual switches within servers.

The devices which add VID to untagged frames need to be informed of the local VID. If data frames from VMs already have VID encoded in data frames, then there has to be a mechanism to notify the first switch port facing the VMs to convert the VID encoded by the VMs to the local VID which is assigned for the virtual network under the new NVE. That means when a VM is moved to a new location, its immediate adjacent switch port has to be informed of local VID to convert the VID encoded in the data frames from the VM.

NVE will need the mapping between local VID and the VNID to be used to face L3 underlay network.

#### 4.1.1. Local VIDs Managed by External Controller

Most likely the VM assignment to a physical location is managed by a non-networking entity, e.g. VM Manager or a Server Manager. NVEs may not be aware of VMs being added or deleted unless NVEs have a north bound interface to a controller which can communicate with VM/server Manager(s).

When NVE can be informed of VMs being added/deleted and their associated tenant virtual networks via its controller, NVE should be able to get the specific VNID from its controller for untagged data frames arriving at its Virtual Access Points [VNo3-framework 3.1.1].

Since local VIDs under each NVE are really locally significant, it might be less confusing to egress NVE if ingress NVE remove the local VID attached to the data frame. So that egress NVE always has to assign its own local VID to data frame before sending the decapsulated data frame to attached VMs.

If, for whatever reason, it is necessary to have local VID in the data frames before encapsulating outer header of EgressNVE-DA/IngressNVE-SA /VNID, NVE should get the specific local VID from the external Controller for those untagged data frames coming to each Virtual Access Point.

If the data frame is tagged before reaching the NVE's Virtual Access Point (e.g. tagged data frames from VMs) and NVE is more than one hop away from VMs, the first (virtual) port facing the VMs has be informed by the external controller of the new local VID to replace the VID encoded in the data frames. For reverse direction, i.e. data frames coming from core towards VMs, the first switching port facing VMs have to convert the VIDs encoded in the data frames to the VIDs used by VMs.

The IEEE802.1Qbg's VDP protocol (Virtual Station Interface (VSI) discovery and configuration protocol) requires hypervisor to send VM profile upon a new VM is instantiated. However, not all hypervisors support this function.

#### 4.1.2. Local VIDs Managed by NVE

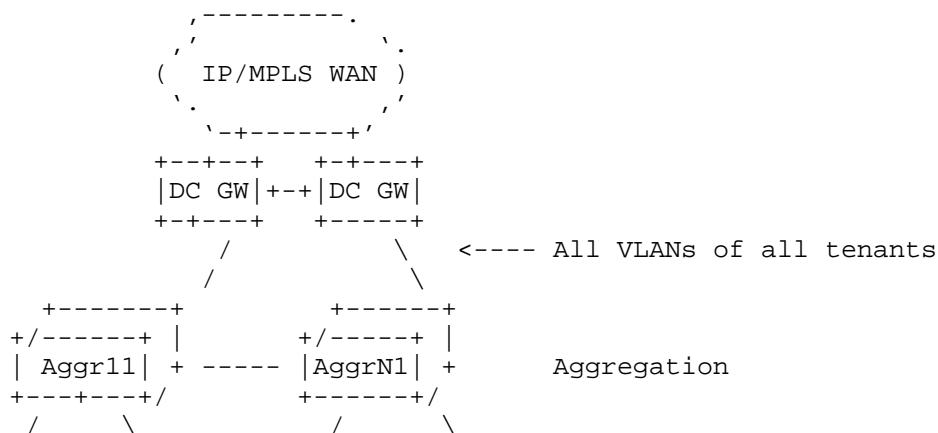
If NVEs don't have interface to any controllers which can be informed of VMs being added to or deleted from NVEs, then NVEs have to learn new VMs/VLANs being attached, figure out to which tenant virtual network those VMs/VLANs belong, and/or age out VMs/VLANs after a specified timer expires. Network management system has to assist NVEs in making the decision, even if the network management system doesn't have interface to VM/server managers.

When NVE receives a data frame with a new VM address (e.g. MAC) in a tagged data frame from its Virtual Access Point, the new VM could be from an existing local virtual network, from a different virtual network (being brought in as the VM being added in), or from an illegal VM.

Upon NVE learns a new VM being added, either by learning a new MAC address or a new VID, it needs its management system to confirm the validity of the new VID and/or new address. If the new address or VID is from invalid or illegal source, the data frame has to be dropped.

#### 4.2. Tenant Virtual Network separation at the physical gateway routers

When a VM communicates with peers in a different subnets, data frames will be sent to the tenant logical Router (Rtr1 or Rtr2 in the Figure 1). Very often, the logical routers of all tenants in a data center are just logical entities (e.g. VRF) on the gateway router(s). That means that all the VLANs for all tenants will be terminated at the Data Center Gateway router(s), as shown in the figure below.





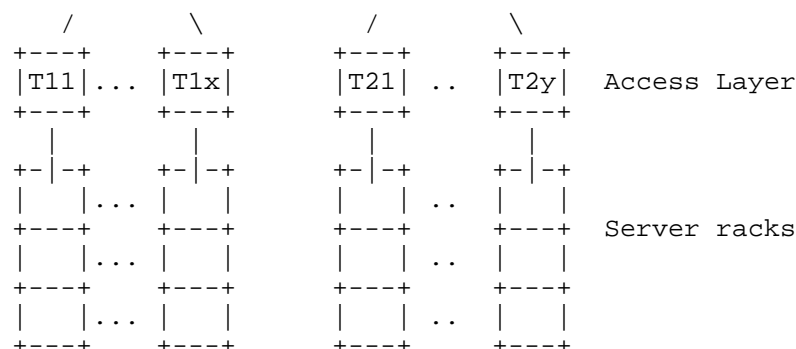


Figure 5: Data Center Physical topology

Gateway routers can mitigate the overwhelming number of virtual network instances by integrating NVE function within the router(s). That requires routers to map VNID to VRF directly if routers' outbound to external network is VPN based. That requires routers to support tens of thousands of VRF instances, which can be challenging to routers.

Data center can also use multiple gateway routers, with each handling a subset of tenants in data centers. That means that each tenant's VMs are only reachable by their designated routers or router ports. With the typical DC design shown in Figure 5, the number of server racks reachable by each gateway router is limited by the number of router ports enabled for the tenant virtual networks. That means the range of locations where each tenant's VMs can be moved across are limited.

When VMs in data center communicates with external peers, data frames have to go through gateway. Even though majority of data centers have much more east west traffic volume than north south traffic volume, majority (as high as 90%) of applications (hosted on servers or VMs) in a data center still communicate with external peers. Just the volume of north south traffic is much less in many data centers.

## 5. Summary and Recommendations

Overlay network can hide individual VMs addresses, making switches/routers in the core scalable. However overlay introduces other challenges, especially when VMs move across wide range of NVEs. This draft is to identify those issues introduced by mobility in

overlay environment, to ensure that they will be addressed by future solutions.

#### 6. Manageability Considerations

#### 7. Security Considerations

Security will be addressed in a separate document.

#### 8. IANA Considerations

None.

#### 9. Acknowledgments

We want to acknowledge the following people for their valuable comments to this draft: David Black, Ben MackCrane, Peter AshwoodSmith, Lucy Yong and Young Lee.

This document was prepared using 2-Word-v2.0.template.dot.

#### 10. References

[NVo3-Problem] Narten, et al, "Problem Statement: Overlays for Network Virtualization." Draft-narten-nvo3-overlay-problem-statement-02, June 2012.

[NVo3-framework] Lasserre, et al, "Framework for DC Network Virtualization". Draft-lasserre-nvo3-framework-02, June 2012

[IEEE802.1Qbg] "MAC Bridges and Virtual Bridged Local Area Networks - Edge Virtual Switch". IEEE802.1Qbg/D2.2, Feb, 2012. Work in progress

[ARMD-Problem] Narten, et al "draft-ietf-armd-problem-statement" in progress, Oct 2011.

[ARMD-Multicast] McBride, Lui, "draft-mcbride-armd-mcast-overview-01", in progress, March 10, 2012

[Gratuitous ARP] S. Cheshire, "IPv4 Address Conflict Detection", RFC 5227, July 2008.

## Authors' Addresses

Linda Dunbar  
Huawei Technologies  
5340 Legacy Drive, Suite 175  
Plano, TX 75024, USA  
Phone: (469) 277 5840  
Email: ldunbar@huawei.com

## Intellectual Property Statement

The IETF Trust takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in any IETF Document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Copies of Intellectual Property disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement any standard or specification contained in an IETF Document. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE

ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS  
FOR A PARTICULAR PURPOSE.

#### Acknowledgment

Funding for the RFC Editor function is currently provided by the  
Internet Society.



Network Working Group  
Internet Draft  
Intended status: Informational  
Expires: December 12, 2012

Maria Napierala  
AT&T  
Luyuan Fang  
Dennis Cai  
Cisco Systems

June 12, 2012

IP-VPN Data Center Problem Statement and Requirements  
draft-fang-vpn4dc-problem-statement-01.txt

Abstract

Network Service Providers commonly use BGP/MPLS VPNs [RFC 4364] as the control plane for virtual networks. This technology has proven to scale to a large number of VPNs and attachment points, and it is well suited for Data Center connectivity, especially when supporting all IP applications.

The Data Center environment presents new challenges and imposes additional requirements to IP VPN technologies, including multi-tenancy support, high scalability, VM mobility, security, and orchestration. This document describes the problems and defines the new requirements.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

This Internet-Draft will expire on December 12, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

Napierala, Fang, Cai    Expire December 12, 2012

[Page 1]

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	IP-VPN in Data Center Network	4
3.1.	Data Center Connectivity Scenarios	5
4.	Data Center Virtualization Requirements	6
5.	Decoupling of Virtualized Networking from Physical Infrastructure	6
6.	Encapsulation/Decapsulation Device for Virtual Network Payloads	7
7.	Decoupling of Layer 3 Virtualization from Layer 2 Topology	8
8.	Requirements for Optimal Forwarding of Data Center Traffic	9
9.	Virtual Network Provisioning Requirements	9
10.	Application of BGP/MPLS VPN Technology to Data Center Network	10
10.1.	Data Center Transport Network	12
10.2.	BGP Requirements in a Data Center Environment	12
11.	Virtual Machine Migration Requirement	14
12.	IP-VPN Data Center Use Case: Virtualization of Mobile Network	15
13.	Security Considerations	17
14.	IANA Considerations	17
15.	Normative References	17
16.	Informative References	17
17.	Authors' Addresses	17
18.	Acknowledgements	18

## Requirements Language

Although this document is not a protocol specification, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC 2119].

## 1. Introduction

Data Centers are increasingly being consolidated and outsourced in an effort, both to improve the deployment time of applications as well as reduce operational costs. This coincides with an increasing demand for compute, storage, and network resources from applications. In order to scale compute, storage, and network resources, physical resources are being abstracted from their logical representation. This is referred as server, storage, and network virtualization. Virtualization can be implemented in various layers of computer systems or networks.

The compute loads of many different customers are executed over a common infrastructure. Compute nodes are often executed as Virtual Machines in an "Infrastructure as a Service" (IaaS) Data Center. The set of virtual machines corresponding to a particular customer should be constrained to a private network.

New network requirements are presented due to the consolidation and virtualization of Data Center resources, public, private, or hybrid. Large scale server virtualization (i.e., IaaS) requires scalable and robust Layer 3 network support. It also requires scalable local and global load balancing. This creates several new problems for network connectivity, namely elasticity, location independence (referred to also as Virtual Machine mobility), and extremely large number of virtual resources.

In the Data Center networks, the VMs of a specific customer or application are often configured to belong to the same IP subnet. Many solutions proposed for large Data Center networks rely on the assumption that the layer-2 inter-server connectivity is required, especially to support VM mobility within a virtual IP subnet. Given that VM mobility consists in moving VMs anywhere within (and even across) Data Centers, the virtual subnet locality associated with small scale deployments cannot be preserved. A Data Center solution should not prevent grouping of virtual resources into IP subnets but the virtual subnets have no benefits of locality across a large data-center.

While some applications may expect to find other peers in a particular user defined IP subnet, this does not imply the need to provide a Layer 2 service that preserves MAC addresses. A network virtualization solution should be able to provide IP unicast connectivity between hosts in the same and different subnets without any assumptions regarding the underlying media layer. A solution should also be able to provide a multicast service that implements IP subnet broadcast as well as IP multicast.



One of the main goals in designing a Data Center network is to minimize the cost and complexity of its core/"fabric" network. The cost and complexity of Data Center network is a function of the number of virtualized resources, that is, the number of "closed user-groups". Data Centers use VPNs to isolate compute resources associated with a specific "closed user-group". Some use VLANs as a VPN technology, others use Layer 3 based solutions often with proprietary control planes. Service Providers are interested in interoperability and in openly documented protocols rather than in proprietary solutions.

## 2. Terminology

AS	Autonomous Systems
DC	Data Center
DCI	Data Center Interconnect
EPC	Evolved Packet Core
End-System	A device where Guest OS and Host OS/Hypervisor reside
IaaS	Infrastructure as a Service
LTE	Long Term Evolution
PCEF	Policy Charging and Enforcement Function
RT	Route Target
ToR	Top-of-Rack switch
VM	Virtual Machine
Hypervisor	Virtual Machine Manager
SDN	Software Defined Network
VPN	Virtual Private Network

## 3. IP-VPN in Data Center Network

In this document, we define the problem statement and requirements for Data Center connectivity based on the assumption that applications require IP connectivity but no Layer 2 direct adjacencies. Applications do not send or receive Ethernet frames directly. They are restricted to IP services due to several reasons such as privileges, address discovery, portability, APIs, etc. IP service can be unicast, VPN broadcast, or multicast.

An IP-VPN DC solution is meant to address IP-only Data Center, defined by a Data Center where VMs, applications, and appliances require only IP connectivity and the underlying DC core infrastructure is IP only. Non-IP applications are addressed by other solutions and are not in scope of this document.

It is also assumed that both IPv4 and IPv6 unicast communication is to be supported. Furthermore, the multicast transmission, i.e., allowing IP applications to send packets to a group of IP addresses should also be supported. The most typical multicast applications are service, network, device discovery applications and content

distribution. While there are simpler and more effective ways to provide discovery services or reliable content delivery, a Data Center solution should support multicast transmission to applications. A Data Center solution should cover the case where the Data Center transport network does not support IP multicast transmission service.

The Data Center multicast service should also support a delivery of traffic to all endpoints of a given VPN even if those endpoints have not sent any control messages indicating the need to receive that traffic. In other words, the multicast service should be capable of delivering the IP broadcast traffic in a virtual topology.

### 3.1. Data Center Connectivity Scenarios

There are three different cases of Data Center (DC) network connectivity:

1. Intra-DC connectivity: Private network connectivity between compute resources within a public (or private) Data Center.
2. Inter-DC connectivity: Private network connectivity between different Data Centers, either public or private.
3. Client-to-DC connectivity: Connectivity between client and a private or public Data Center. The later includes interconnection between a service provider and a public Data Center (which may belong to the same or different service provider).

Private network connectivity within the Data Center requires network virtualization solution. In this document we define Layer 3 VPN requirements to Data Center network virtualization. The Layer 3 VPN technology (i.e., MPLS/BGP VPN) also applies to the interconnection of different data-centers.

When private networks interconnect with public Data Centers, the VPN provider must interconnect with the public Data Center provider. In this case we are in the presence of inter-provider VPNs. The Inter-AS MPLS/BGP VPN Options A, B, or C [RFC 4364] provide network-to-network interconnection service and they constitute the basis of SP network to public Data Center network connectivity. There might incremental improvements to the existing inter-AS solutions, pertaining to scalability and security, for example.

Service Providers can leverage their existing Layer 3 VPN services and provide private VPN access from client's branch sites to client's own private Data Center or to SP's own Data Center. The service provider-based VPN access can provide additional value compared with public internet access, such as security, QoS, OAM, and troubleshooting.

#### 4. Data Center Virtualization Requirements

Private network connection service in a Data Center must provide traffic isolation between different virtual instances that share a common physical infrastructure. A collection of compute resources dedicated to a process or application is referred to as a "closed user-group". Each "closed user-group" is a VPN in the terminology used by IP VPNs.

Any DC solution needs to assure network isolation among tenants or applications sharing the same Data Center physical resources. A DC solution should allow a VM or application end-point to belong to multiple closed user-groups/VPNs. A closed user-group should be able to communicate with other closed-user groups according to specified routing policies. A customer or tenant should be able to define multiple closed user-groups.

Typically VPNs that belong to different tenants do not communicate with each other directly but they should be allowed to access common appliances such as storage, database services, security services, etc. It is also common for tenants to deploy a VPN per "application tier" (e.g. a VPN for web front-ends and a different VPN for the logic tier). In that scenario most of the traffic crosses VPN boundaries. That is also the case when "network attached storage" (NAS) is used or when databases are deployed as-a-service.

Another reason for the Data Center network virtualization is the need to support VM move. Since the IP addresses used for communication within or between applications may be anywhere across the data-center, using a virtual topology is an effective way to solve this problem.

#### 5. Decoupling of Virtualized Networking from Physical Infrastructure

The Data Center switching infrastructure (access, aggregation, and core switches) should not maintain any information that pertains to the virtual networks. Decoupling of virtualized networking from the physical infrastructure has the following advantages: 1) provides

better scalability; 2) simplifies the design and operation; 3) reduces the cost of a Data Center network. It has been proven (in Internet and in large BGP IP VPN deployments) that moving complexity associated with virtual entities to network edge while keeping network core simple has very good scaling properties.

There should be a total separation between the virtualized segments (virtual network interfaces that are associated with VMs) and the physical network (i.e., physical interfaces that are associated with the data-center switching infrastructure). This separation should include the separation of the virtual network IP address space from the physical network IP address space. The physical infrastructure addresses should be routable in the underlying Data Center transport network, while the virtual network addresses should be routable on the VPN network only. Not only should the virtual network data plane be fully decoupled from the physical network, but its control plane should be decoupled as well. In order to decouple virtual and physical networks, the virtual networking should be treated as an "infrastructure" application. Only the solutions that meet those requirements would provide a truly scalable virtual networking.

MPLS labels provide the necessary information to implement VPNs. When crossing the Data Center infrastructure the virtual network payloads should be encapsulated in IP or GRE [RFC 4023], or native MPLS envelopes.

## 6. Encapsulation/Decapsulation Device for Virtual Network Payloads

In order to scale a virtualized Data Center infrastructure, the encapsulation (and decapsulation) of virtual network payloads should be implemented on a device as close to virtualized resources as possible. Since the hypervisors in the end-systems are the devices at the edge of a Data Center network they are the most optimal location for the VPN encap/decap functionality. Data-plane device that implements the VPN encap/decap functionality acts as the first-hop router in the virtual topology.

The IP-VPN solution for Data Center should also support deployments where it is not possible or not desirable to implement VPN encapsulation in the hypervisor/Host OS. In such deployments encap/decap functionality may be implemented in an external physical switch such as aggregation switch or top-of-rack switch. The external device implementing VPN tunneling functionality should be as close as possible to the end-system itself. The same DC solution should support deployments with both, internal (in a hypervisor) and external (outside of a hypervisor) encap/decap devices.

Whenever the VPN forwarding functionality (i.e., the data-plane device that encapsulates packets into, e.g., MPLS-over-GRE header) is implemented in an external device, the VPN service itself must be delivered to the virtual interfaces visible to the guest OS. However, the switching elements connecting the end-system to the encap/decap device should not be aware of the virtual topology. Instead, the VPN endpoint membership information might be, for example, communicated by the end-system using a signaling protocol. Furthermore, for an all-IP solution, the Layer 2 switching elements connecting the end-system to the encap/decap device should have no knowledge of the VM/application endpoints. In particular, the MAC addresses known to the guest OS should not appear on the wire.

## 7. Decoupling of Layer 3 Virtualization from Layer 2 Topology

The IP-VPN approach to Data Center network design dictates that the virtualized communication should be routed, not bridged. The Layer 3 virtualization solution should be decoupled from the Layer 2 topology. Thus, there should be no dependency on VLANs or Layer 2 broadcast.

In solutions that depend on Layer 2 broadcast domains, the VM-to-VM communication is established based on flooding and data plane MAC learning. Layer 2 MAC information has to be maintained on every switch where a given VLAN is present. Even if some solutions are able to eliminate data plane MAC learning and/or unicast flooding across Data Center core network, they still rely on VM MAC learning at the network edge and on maintaining the VM MAC addresses on every (edge) switch where the Layer 2 VPN is present.

The MAC addresses known to guest OS in end-system are not relevant to IP services and introduce unnecessary overhead. Hence, the MAC addresses associated with virtual machines should not be used in the virtual Layer 3 networks. Rather, only what is significant to IP communication, namely the IP addresses of the VMs and application endpoints should be maintained by the virtual networks. An IP-VPN solution should forwards VM traffic based on their IP addresses and not on their MAC addresses.

From a Layer 3 virtual network perspective, IP packets should reach the first-hop router in one-hop, regardless of whether the first-hop router is a hypervisor/Host OS or it is an external device. The VPN first-hop router should always perform an IP lookup on every packet it receives from a VM or an application. The first-hop router should encapsulate the packets and route them towards the destination end-system. Every IP packet should be forwarded along the shortest path towards a destination host or appliance,

regardless of whether the packet's source and destination are in the same or different subnets.

## 8. Requirements for Optimal Forwarding of Data Center Traffic

The Data Center solutions that optimize for the maximum utilization of compute and storage resources require that those resources may be located anywhere in the data-center. The physical and logical spreading of appliances and computations implies a very significant increase in data-center infrastructure bandwidth consumption. Hence, it is important that DC solutions are efficient in terms of traffic forwarding and assure that packets traverse Data Center switching infrastructure only once. This is not possible in DC solutions where a virtual network boundary between bridging (Layer 2) and routing (Layer 3) exists anywhere within the Data Center transport network. If a VM can be placed in an arbitrary location, mixing of the Layer 2 and the Layer 3 solutions may cause the VM traffic traverse the Data Center core multiple times before reaching the destination host.

It must be also possible to send the traffic directly from one VM to another VM (within or between subnets) without traversing through a midpoint router. This is important given that most of the traffic in a Data Center is within the VPNs.

## 9. Virtual Network Provisioning Requirements

IP-VPN DC has to provide fast and secure provisioning (with low operational complexity) of VPN connectivity for a VM within a Data Center and across Data Centers. This includes interconnecting VMs within and across physical Data Centers in the context of a virtual networking. It also includes the ability to connect a VM to a customer VPN outside the Data Center, thus requiring the ability to provision the communication path within the Data Center to the customer VPN.

The VM provisioning should be performed by an orchestration system. The orchestration system should have a notion of a closer user-group/tenant and the information about the services the tenant is allowed to access. The orchestration system should allocate an IP address to a VM. When the VM is provisioned, its IP address and the closed user-group/VPN identifier (VPN-ID) should be communicated to the host OS on the end-system. There should a centralized database system (possibly with a distributed implementation) that will contain the provisioning information regarding VPN-IDs and the services the corresponding VPNs could

access. This information should be accessible to the virtual network control plane.

The orchestration system should be able to support the specification of fine grain forwarding policies (such as filtering, redirection, rate limiting) to be injected as the traffic flow rules into the virtual network.

Common APIs can be a simple and a useful step to facilitate the provisioning processes. Authentication is required when a VM is being provisioned to join an IP VPN.

An IP-VPN Data Center networking solution should seamlessly support VM connectivity to other network devices (such as service appliances or routers) that use the traditional BGP/MPLS VPN technology.

#### 10. Application of BGP/MPLS VPN Technology to Data Center Network

BGP IP VPN technologies (based on [RFC 4364]) have proven to be able to scale to a large number of VPNs (tens of thousands) and customer routes (millions) while providing for aggregated management capability. Data Center networks could use the same transport mechanisms as used today in many Service Provider networks, specifically the MPLS/BGP VPNs that often overlay huge transport areas.

MPLS/BGP VPNs use BGP as a signaling protocol to exchange VPN routes. IP-VPN DC solution should consider that it might not be feasible to run BGP protocol on a hypervisor or external switch such as top-of-rack. This includes functions like BGP route selection and processing of routing policies, as well as handling MP-BGP structures like Route Distinguishers and Route Targets. Rather, it might be preferable to use a signaling mechanism that is more familiar and compatible with the methods used in the application software development. While network devices (such as routers and appliances) may choose to receive VPN signaling information directly via BGP, the end-systems/switches may choose other type of interface or protocol to exchange virtual end-point information. The IP VPN solution for Data Center should specify the mapping between the signaling messages used by the hypervisors/switches and the MP-BGP routes used by MP-BGP speakers participating in the virtual network.

In traditional WAN deployments of BGP IP VPNs [RFC 4364], the forwarding function and control function of a Provider Edge (PE) device have co-existed within a single physical router. In a Data Center network, the PE plays a role of the first-hop router, in a

virtual domain. The signaling exchanged between forwarding and control planes in a PE has been proprietary to a specific PE router/vendor. When BGP IP VPNs are applied to a Data Center network, the signaling used between the control plane and forwarding should be open to provisioning and standardization. We explore this requirement in more detail below.

When MPLS/BGP VPNs [RFC 4364] are used to connect VMs or application endpoints, it might be desirable for a hypervisor's host or an external switch (such as TOR) to support only the forwarding aspect of a Provider Edge (PE) function. The VMs or applications would act as Customer Edges (CEs) and the virtual networks interfaces associated with the VMs/applications as CE interfaces. More specifically, a hypervisor/first-hop switch would support only the creation and population of VRF tables that store the forwarding information to the VMs and applications. The forwarding information should include 20-bit label associated with a virtual interface (i.e., a specific VM/application endpoint) and assigned by the destination PE. This label has only a local significance within a destination PE. A hypervisor/first-hop switch would not need to support BGP, a protocol familiar to network devices.

When a PE forwarding function is implemented on an external switch, such as aggregation or top-of-rack switch, the end-system must be able to communicate the endpoint and its VPN membership information to the external switch. It should be able to convey the endpoint's instantiation as well as removal events.

An IP-VPN Data Center networking solution should be able to support a mixture of internal PEs (implemented in hypervisors/Host OS) and external PEs (implemented on external to the end-system devices).

The IP-VPN DC solution should allow BGP/MPLS VPN-capable network devices, such as routers or appliances, to participate directly in a virtual network with the Virtual Machines and applications. Those network devices can participate in isolated collections of VMs, i.e., in isolated VPNs, as well as in overlapping VPNs (called "extranets" in BGP/MPLS VPN terminology).

The device performing PE forwarding function should be capable of supporting multiple Virtual Routing and Forwarding (VRF) tables representing distinct "close user groups". It should also be able to associate a virtual interface (corresponding to a VM or application endpoint) with a specific VRF.

The first-hop router has to be capable of encapsulating outgoing traffic (end-system towards Data Center network) in IP/GRE or MPLS envelopes, including the per-prefix 20-bit VPN label. The first-hop router has to be also capable of associating incoming packets from



a Data Center network with a virtual interface, based on the 20-bit VPN label contained in the packets.

The protocol used by the VPN first-hop routers to signal VPNs should be independent of the transport network protocol as long as the transport encapsulation has the ability to carry a 20-bit VPN label.

### 10.1. Data Center Transport Network

MPLS/VPN technology based on [RFC 4364] specifies several different encapsulation methods for connecting PE routers, namely Label Switched Paths (LSPs), IP tunneling, and GRE tunneling. If LSPs are used in the transport network they could be signaled with LDP, in which case host (/32) routes to all PE routers must be propagated throughout the network, or with RSVP-TE, in which case a full mesh of RSVP-TE tunnels is required, generating a lot of state in the network core. If the number of LSPs is expected to be high, due to a large size of Data Center network, then IP or GRE encapsulation can be used, where the above mentioned scalability is not a concern due to route aggregation property of IP protocols.

### 10.2. BGP Requirements in a Data Center Environment

#### 10.2.1. BGP Convergence and Routing Consistency

BGP was designed to carry very large amount of routing information but it is not a very fast converging protocol. In addition, the routing protocols, including BGP, have traditionally favored convergence (i.e., responsiveness to route change due to failure or policy change) over routing consistency. Routing consistency means that a router forwards a packet strictly along the path adopted by the upstream routers. When responsiveness is favored, a router applies a received update immediately to its forwarding table before propagating the update to other routers, including those that potentially depend upon the outcome of the update. The route change responsiveness comes at the cost of routing blackholes and loops.

Routing consistency across Data Center is important because in large Data Centers thousands of Virtual Machines can be simultaneously moved between server racks due to maintenance, for example. If packets sent by the Virtual Machines that are being moved are dropped (because they do not follow a live path), the active network connections on those VMs will be dropped. To minimize the disruption to the established communications during VM migration, the live path continuity is required.

### 10.2.2. VM Mobility Support

To overcome BGP convergence and route consistency limitations, the forwarding plane techniques that support fast convergence should be used. In fact, there exist forwarding plane techniques that support fast convergence by removing from the forwarding table a locally learned route and instantaneously using already installed new routing information to a given destination. This technique is often referred to as "local repair". It allows to forward traffic (almost) continuously to a VM that has migrated to a new physical location using an indirect forwarding path or tunnel via VM's old location (i.e., old VM forwarder). The traffic path is restored locally at the VM's old location while the network converges to the new location of the migrated VM. Eventually, the network converges to optimal path and bypasses the local repair. BGP should assist in the local repair techniques by advertizing multiple and not only the best path to a given destination.

### 10.2.3. Optimizing Route Distribution

When virtual networks are triggered based on the IP communication (as proposed in this document), the Route Target Constraint extension [RFC 4684] of BGP should be used to optimize the route distribution for sparse virtual network events. This technique ensures that only those VPN forwarders that have local participants in a particular data plane event receive its routing information. This also decreases the total load on the upstream BGP speakers.

### 10.2.4. Inter-operability with MPLS/BGP VPNs

As was stated in section 10, the IP-VPN DC solution should be fully inter-operable with MPLS/BGP VPNs. MPLS/BGP VPN technology is widely supported on routers and other appliances. When connecting a Data Center virtual network with other services/networks, it is not necessary to advertize the specific VM host routes but rather the aggregated routing information. A router or appliance within a Data Center can be used to aggregate VPN's IP routing information and advertize the aggregated prefixes. The aggregated prefixes would be advertized with the router/appliance IP address as BGP next-hop and with locally assigned aggregate 20-bit label. The aggregate label will trigger a destination IP lookup in its corresponding VRF on all the packets entering the virtual network.

## 11. Virtual Machine Migration Requirement

The "Virtual Machine live migration" (a.k.a. VM mobility) is highly desirable for many reasons such as efficient and flexible resource sharing, Data Center migration, disaster recovery, server redundancy, or service bursting. VM live migration consists in moving a virtual machine from one physical server to another, while preserving the VM's active network connections (e.g., TCP and higher-level sessions).

VM live mobility primarily happens within the same physical Data Center but VM live mobility between Data Centers might be also required. The IP-VPN Data Center solutions need to address both intra-Data Center and inter-Data Center VM live mobility.

Traditional Data Center deployments have followed IP subnet boundary, i.e., hosts often stayed in the same IP subnet and a host had to change its IP address when it moved to a different location. Such architecture have worked well when hosts were dedicated to an application and resided in physical proximity to each other. These assumptions are not true in the IaaS environment where compute resources associated with a given application can be spread and dynamically move across a large Data Center.

Many DC design proposals are trying to address the VM mobility with data-center wide VLANs using Data Center-wide Layer 2 broadcast domains. With data-center wide VLANs, a VM move is handled by generating gratuitous ARP reply to update all ARP caches and switch learning tables. Since a virtual subnet locality cannot be preserved in a large Data Center, a virtual subnet (VLAN) must be present on every Data Center switch, limiting the number of virtual networks to 4094. Even if a Layer 2 Data Center solution is able to minimize or eliminate the ARP flooding across Data Center core, all edge switches still have to perform dynamic VM MAC learning and maintain VM's MAC-to-IP mappings.

Since in large Data Centers physical proximity of computing resources cannot be assumed, grouping of hosts into subnets does not provide any VM mobility benefits. Rather, VM mobility in a large Data Center should be based on a collection of host routes spread randomly across a large physical area.

When dealing with IP-only applications it is not only sufficient but optimal to forward the traffic based on Layer 3 rather than on Layer 2 information. The MAC addresses of Virtual Machines are irrelevant to IP services and introduce unnecessary overhead (i.e., maintaining ARP caches of VM MACs) and complications when VMs move (e.g., when VM's MAC address is changed in its new location). IP-based VPN connectivity solution is a cost effective and scalable approach to

solve VM mobility problem. In IP-VPN DC a VM move is handled by a route advertisement.

To accommodate live migration of Virtual Machines, it is desirable to assign a permanent IP address to a VM that remains with the VM after it moves. Typically, a VM/application reaches the off-subnet destinations via a default gateway, which should be the first-hop router (in the virtual topology). A VM/application should reach the on-subnet destinations via an ARP proxy which again should be the VPN first-hop router. A VM/application cannot change the default gateway's IP and MAC addresses during live migration, as it would require changes to TCP/IP stack in the guest OS. Hence, the first-hop VPN router should use a common, locally significant IP address and a common virtual MAC address to support VM live mobility. More specifically, this IP address and the MAC address should be the same on all first-hop VPN routers in order to support the VM moves between different physical machines. Moreover, in order to preserve virtual network and infrastructure separation, the IP and MAC addresses of the first-hop routers should be shared among all virtual IP-subnets/VPNs. Since the first-hop router always performs an IP lookup on every packet destination IP address, the VM traffic is forwarded on the optimal path and traverses the Data Center network only once.

The VM live migration has to be transparent to applications and any external entity interacting with the applications. This implies that the VM's network connectivity restoration time is critical. The transport sessions can typically survive over several seconds of disruption, however, applications may have sub-second latency requirement for their correct operation.

To minimize the disruption to the established communications during VM migration, the control plane of a DC solution should be able to differentiate between VM activation in a new location from advertising its host route to the network. This will enable the VPN first-hop routers forwarders to install a route to VM's new location prior to its migration, allowing the traffic to be tunneled via the first-hop router at the VM's old location. There are techniques available in BGP as well as in forwarding plane that support fast convergence due to withdrawal or replacement of current or less preferred forwarding information (see section 10.2 for more detailed description of such technique).

## 12. IP-VPN Data Center Use Case: Virtualization of Mobile Network

Application access is being done increasingly from clients such as cell phones or tablets connecting via private or public WiFi access

points, or 3G/LTE wireless access. Enterprises with a mobile workforce need to access resources in the enterprise VPN while they are traveling, e.g., sales data from a corporate database. The mobile workforce might also, for security reasons, be equipped with disk-less notebooks which rely on the enterprise VPN for all file accesses. The mobile workforce applications may occasionally need to utilize the compute resources and other functions (e.g., storage) that the enterprise hosts on the infrastructure of a cloud computing provider. The mobile devices might require simultaneous access to resources in both, the cloud infrastructure as well as the enterprise VPN.

The enterprise wide area network may use a provider-based MPLS/BGP VPN service. The wireless service providers already use MPLS/BGP VPNs for enterprise customer isolation in the mobile packet core elements. Using the same VPN technology in the service provider Data Center network (or in a public Data Center network) is a natural extension.

Furthermore, there is a need to instantiate mobile applications themselves as virtual networks in order to improve application performance (e.g., latency, Quality-of-Service) or to enable new applications with specialized requirements. In addition it might be required that the application's computing resource is made to be part of the mobility network itself and placed as close as possible to a mobile user. Since LTE data and voice applications use IP protocols only, the IP-VPN solution to virtualization of compute resources in mobile networks would be the optimal approach.

The infrastructure of a large scale mobility network could itself be virtualized and made available in the form of virtual private networks to organizations that do not want to spend the required capital. The Mobile Core functions can be realized via software running on virtual machines in a service-provider-class compute environment. The functional entities such as Service-Gateways (S-GW), Packet-Gateways (P-GW), or Policy Charging and Enforcement Function (PCEF) of the LTE system can be run as applications on virtual machines, coordinated by an orchestrator and managed by a hypervisor. Virtualized packet core network elements (PCEF, S-GW, P-GW) could be placed anywhere in the mobile network infrastructure, as long as the IP connectivity is provided. The virtualization of the Mobile Core functions running on a private computing environment has many benefits, including faster service delivery, better economies of scale, simpler operations. Since the LTE (Long Term Evolution) and Evolved Packet Core (EPC) system are all-IP networks, the IP-VPN solution to mobile network virtualization is the best fit.

13. Security Considerations

The document presents the problems need to be addressed in the L3VPN for Data Center space. The requirements and solutions will be documented separately.

The security considerations for general requirements or individual solutions will be documented in the relevant documents.

14. IANA Considerations

This document contains no new IANA considerations.

15. Normative References

[RFC 4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC 4023] Worster, T., Rekhter, Y. and E. Rosen, "Encapsulating in IP or Generic Routing Encapsulation (GRE)", RFC 4023, March 2005.

[RFC 4684] Marques, P., Bonica, R., Fang, L., Martini, L., Raszkuk, R., Patel, K. and J. Guichard, "Constrained Route Distribution for Border Gateway Protocol/Multiprotocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)", RFC 4684, November 2006.

16. Informative References

[RFC 2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

17. Authors' Addresses

Maria Napierala  
AT&T  
200 Laurel Avenue  
Middletown, NJ 07748  
Email: mnapierala@att.com

Luyuan Fang  
Cisco Systems  
111 Wood Avenue South

Iselin, NJ 08830, USA  
Email: lufang@cisco.com

Dennis Cai  
Cisco Systems  
725 Alder Drive  
Milpitas, CA 95035, USA  
Email: dcai@cisco.com

## 18. Acknowledgements

The authors would like to thank Pedro Marques for his helpful comments and input.

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 10, 2013

Y. Gu  
W. Hao  
Huawei  
July 9, 2012

Analysis of external assistance to NVE and consideration of architecture  
draft-gu-nvo3-overlay-cp-arch-00

## Abstract

Draft [overlay-cp] has introduced some control plan requirements and characteristics. From NVE's perspective, this draft describes what assistance is needed to make NVE satisfy the requirements and characteristics introduce in [overlay-cp]. Not all of these assistance is necessarily achieved by an external controller. Some of the assistance requirements can be regarded as a complementarity requirements to [overlay-cp]. while others are requirements to an assistance Database. This draft also provide considerations on how the network virtualization architecture should be like and how these assistance can be fulfilled. The target is to help the working group to figure out the architecture of overlay control plane, instead of providing solutions.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 10, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents



(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminologies and concepts . . . . .	3
3. The fundamental requirements and characteristics . . . . .	5
3.1. Assistance to NVE . . . . .	6
3.1.1. Assistance from TES . . . . .	6
3.2. Access Control List . . . . .	7
3.3. QoS . . . . .	7
3.4. DHCP Snooping . . . . .	7
3.5. NVE to VNI Registration . . . . .	7
3.6. VNI to Multicast Addr Mapping . . . . .	8
3.7. Synchronization . . . . .	8
4. Implementation Options and Architecture considerations . . . . .	8
4.1. Exclusively using External Controller . . . . .	9
4.2. Hybrid of External Controller and Centralized Database . . . . .	10
4.2.1. Brief introduction of VDP profile database and work flow . . . . .	10
4.2.2. Example Architecture and Work Flow . . . . .	12
5. Summary . . . . .	13
6. Security Considerations . . . . .	13
7. References . . . . .	14
7.1. Normative Reference . . . . .	14
7.2. Informative Reference . . . . .	14
Authors' Addresses . . . . .	14

## 1. Introduction

Draft [overlay-cp] has introduced some control plane requirements and characteristics. From NVE's perspective, this draft describes what assistance is needed to make NVE satisfy the requirements and characteristics introduced in [overlay-cp]. Not all of these assistance is necessarily achieved by an external controller. Some of the assistance requirements can be regarded as a complementarity requirements to [overlay-cp]. While others are requirements to an assistance Database. This draft also provides considerations on how the network virtualization architecture should be and how these assistance can be fulfilled. The target is to help the working group to figure out the architecture of overlay control plane, instead of providing solutions.

## 2. Terminologies and concepts

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The document uses terms defined in [framework] and [overlay-cp].

VN: Virtual Network. This is a virtual L2 or L3 domain that belongs to a tenant.

VNI: Virtual Network Instance. This is one instance of a virtual overlay network. Two Virtual Networks are isolated from one another and may use overlapping addresses.

Virtual Network Context or VN Context: Field that is part of the overlay encapsulation header which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE. The egress NVE uses this field to determine the appropriate virtual network context in which to process the packet. This field MAY be an explicit, unique (to the administrative domain) virtual network identifier (VNID) or MAY express the necessary context information in other ways (e.g. a locally significant identifier).

VNID: Virtual Network Identifier. In the case where the VN context has global significance, this is the ID value that is carried in each data packet in the overlay encapsulation that identifies the Virtual Network the packet belongs to.

NVE: Network Virtualization Edge. It is a network entity that sits on the edge of the NVO3 network. It implements network

virtualization functions that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses). An NVE could be implemented as part of a virtual switch within a hypervisor, a physical switch or router, a Network Service Appliance or even be embedded within an End Station.

**Underlay or Underlying Network:** This is the network that provides the connectivity between NVEs. The Underlying Network can be completely unaware of the overlay packets. Addresses within the Underlying Network are also referred to as "outer addresses" because they exist in the outer encapsulation. The Underlying Network can use a completely different protocol (and address family) from that of the overlay.

**Data Center (DC):** A physical complex housing physical servers, network switches and routers, Network Service Appliances and networked storage. The purpose of a Data Center is to provide application and/or compute and/or storage services. One such service is virtualized data center services, also known as Infrastructure as a Service.

**VM: Virtual Machine.** Several Virtual Machines can share the resources of a single physical computer server using the services of a Hypervisor (see below definition).

**Hypervisor:** Server virtualization software running on a physical compute server that hosts Virtual Machines. The hypervisor provides shared compute/memory/storage and network connectivity to the VMs that it hosts. Hypervisors often embed a Virtual Switch (see below).

**Virtual Switch:** A function within a Hypervisor (typically implemented in software) that provides similar services to a physical Ethernet switch. It switches Ethernet frames between VMs' virtual NICs within the same physical server, or between a VM and a physical NIC card connecting the server to a physical Ethernet switch. It also enforces network isolation between VMs that should not communicate with each other.

**Tenant:** A customer who consumes virtualized data center services offered by a cloud service provider. A single tenant may consume one or more Virtual Data Centers hosted by the same cloud service provider.

**Tenant End System:** It defines an end system of a particular tenant, which can be for instance a virtual machine (VM), a non-virtualized server, or a physical appliance.

**Virtual Access Points (VAPs):** Tenant End Systems are connected to the

Tenant Instance through Virtual Access Points (VAPs). The VAPs can be in reality physical ports on a ToR or virtual ports identified through logical interface identifiers (VLANs, internal VSwitch Interface ID leading to a VM).

VN Name: A globally unique name for a VN. The VN Name is not carried in data packets originating from End Stations, but must be mapped into an appropriate VN-ID for a particular encapsulating technology. Using VN Names rather than VN-IDs to identify VNs in configuration files and control protocols increases the portability of a VDC and its associated VNs when moving among different administrative domains (e.g. switching to a different cloud service provider).

VSI: Virtual Station Interface. Typically, a VSI is a virtual NIC connected directly with a VM. [Qbg]

### 3. The fundamental requirements and characteristics

In this section, we make a summary of the fundamental requirements and characteristics made in [overlay-cp].

Summary of requirements:

- o Inner to Outer address mapping
- o Underlying Network Multi-Destination Delivery Address(es)
- o VN Connect/Disconnect Notification
- o VN Name to VN-ID Mapping

Summary of characteristics:

- o As few local caching state as better
- o Fast acquisition of needed state
- o Fast detection/update of stale cached state information
- o Minimize processing overhead
- o Highly scalable
- o Minimize the complexity of the implementation
- o Extensible

- o Simple protocol configuration
- o Do not rely on IP Multicast
- o Flexible mapping sources

### 3.1. Assistance to NVE

In this section, we describe the assistance to NVE as an addition to the requirements enumerated in the above section. Meanwhile the additional requirements must satisfy the required characteristic. We call it assistance, instead of control plane requirements, since the assistance can be achieved by a controller, or a database, which is not traditionally in concept of control plane.

In following section, more than one options to enable these assistance are introduced. No matter what kind of control plane components are finally adopted by the working, the assistance requirements must be satisfied.

#### 3.1.1. Assistance from TES

In draft [tes-nve-mechanism], some requirements and possible mechanisms to enable the requirements are described. These requirements are the assistance that TES can provides, maybe together with external entities, e.g. controllers or profile Database. A summary is enumerated here.

REQUIREMENT-1: The TNP (TES to NVE notification mechanism and protocol) MUST support TES to notify NVE about the VM's status, including but not limited to Start up, Shut down, Emigration and Immigration.

REQUIREMENT-2: The TNP MUST support TES to notify NVE about the VM's VN Clue, which can be one identifier or a combination of several identifier.

REQUIREMENT-3: The TNP MUST support TES to notify NVE about the VM's inner address. The inner address MUST include one or both of MAC address of VM's virtual NIC and VM's IP address. And it SHOULD be extensible to carry new address type.

REQUIREMENT-4: The TNP MUST support NVE to notify TES about the VM's local tag. The local Tag type supported by TNP MUST include IEEE 802.1Q tag. And it SHOULD be extensible to carry other type of local tag.

REQUIREMENT-5: The TNP SHOULD support NVE to notify TES about the VM's traffic PCP value.

The following sections are the assistance the NVE needs but can be provided by entities other than TES, e.g. by an external controller or a database. These assistance requirements are complementary to those introduced in . [overlay-cp]

### 3.2. Access Control List

While VAP identify the a new membership, be a VM or a physical server, NVE needs to get the Access Control List to the member. The ACL maybe associate with a specific member or associate with a specific VNI. If the ACL is associate with a specific VNI, NVE only needs to get the ACL at the first time the NVE is associate with the VNI.

If the ACL changes, e.g. rules change or deleting, the assistance subject must be able to notify NVE to update the ACL.

While the member migrates to a new NVE, the NVE must be able to get the ACL as soon as possible.

### 3.3. QoS

Similar to ACL, NVE needs to get the QoS policies while a new member is associated with the NVE. In order to achieve QoS policies, not only the NVE but also the network devices on traffic path other than NVE need to be aware of the QoS policies. But in the NVO3 working group, we only focus on NVE.

While the member migrates to a new NVE, the NVE must be able to get the QoS policies as soon as possible.

### 3.4. DHCP Snooping

While DHCP Snooping function is enabled on NVE, a DHCP snooping table item is created by the access NVE. While VM migrates to a new NVE, the VM may not resend a DHCP request since the migration is transparent to the VM and the IP address must be the same. In this case, the new NVE must be able to get the DHCP Snooping information created by the original NVE by some way. And the original NVE must be able to delete the DHCP Snooping information timely.

### 3.5. NVE to VNI Registration

While the first membership to a specific VNI is created on NVE, NVE need to register the association to an external entity. The reason

for this is to enable an a global view of which NVEs belongs to a specific VNI. Every NVE must be aware of NVE to VNI mapping for multicast in a single VNI or to update the QoS/ACL policies. For example, all NVEs responsible to at least one member belong to a particular VNI have to be notified of updated ACL or QoS policies related to this VNI.

### 3.6. VNI to Multicast Addr Mapping

NVE can get the inner to outer address mapping through control plane assistance or through data plane learning. In the case of latter, NVE must be able to learn the VNI to Multicast address mapping in order to forward unknown unicast and broadcast traffic.

### 3.7. Synchronization

This assistance a general requirement. For whatever information NVE get from external entity, while the origin of the information is changing, all relevant NVE who have local copy of the information must be able to synchronize with the origin. Some examples of the information are ACL, QoS, Inner to Outer address mapping, VN Name to VNID mapping, and NVEs to VNI global view.

## 4. Implementation Options and Architecture considerations

The combination of requirements in Section 3 and Section 4 are the assistance that NVE need in order to fulfill the overlay forwarding in a way satisfying the characteristic in Section 3. Not all of the assistance is necessarily regarded as requirements to an external controller. In fact, there are more than one way to enable these requirements. In this section, we introduce 2 kinds of assistance subject to enable the above requirements. These should not be regarded as solution proposals, but considerations on overlay control plan components.

In this draft, we only consider the situation where external NVE is embedded on network devices and VMs access to NVE via hypervisor. But for other cases, the mechanism introduced here can also be used, with necessary prune.

Two assistance subjects are introduced, including external controller and centralized database. It's not feasible to use only database, e.g. it's hard for database to synchronize mapping and QoS/ACL polices among all VNI-relevant NVEs. But a centralized database can offload much work from controller.

## 4.1. Exclusively using External Controller

Only an external controller is used to assist NVE for virtualization network forwarding. The controller might have a database on it or directly attached.

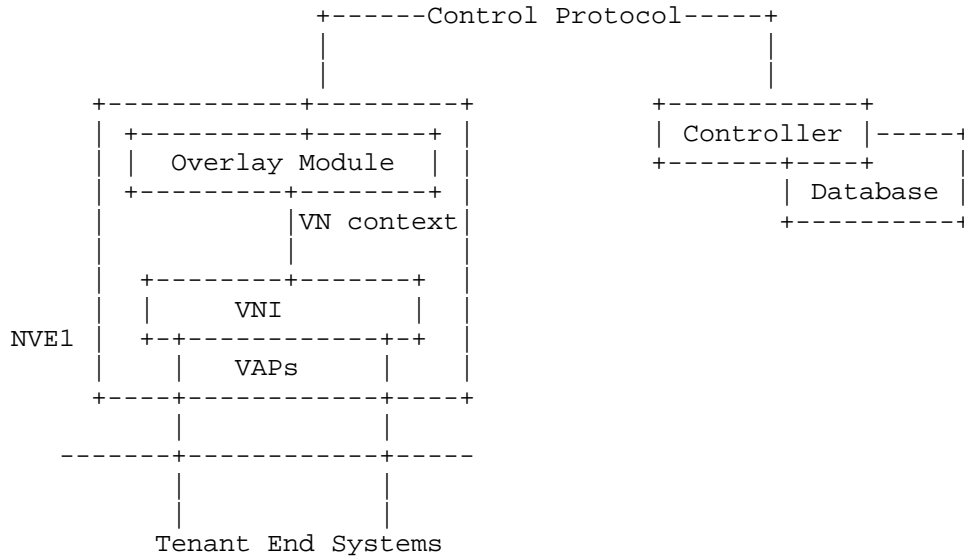


Fig1. Architecture with only controller

The working flow is as follows.

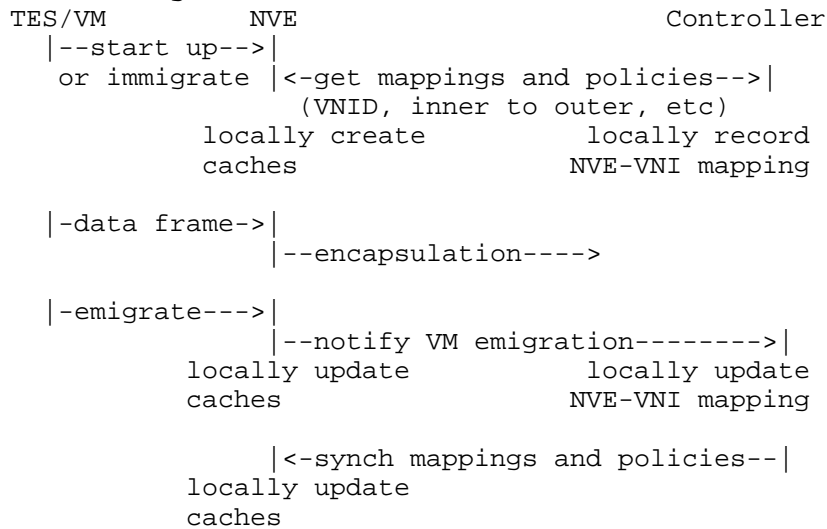


Fig2. Work flow with controller assists NVE



## 4.2. Hybrid of External Controller and Centralized Database

### 4.2.1. Brief introduction of VDP profile database and work flow

Take Profile Database introduced in IEEE 802.1Qbg as an example of the Centralized Database. In IEEE 802.1Qbg, a database is mentioned on how to assist the VDP protocol. It's not standardized in IEEE 802.1Qbg, but is a fundamental knowledge while VDP is defined. Please refer to to find out the brief protocol introduction of VDP. The following figure shows what is profile database and how it works. [tes-nve-mechanism]

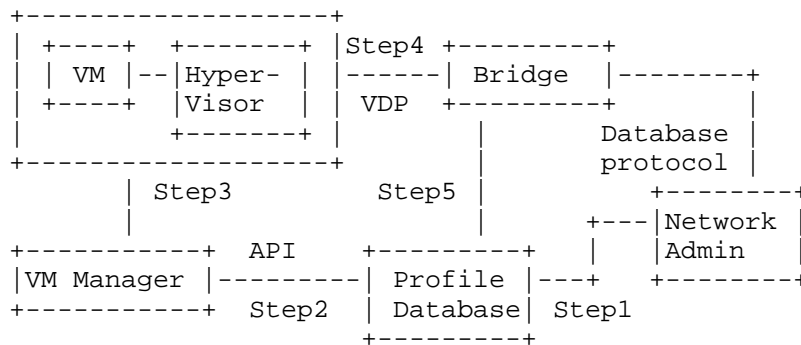


Fig3. VDP Profile Database

A profile database is a centralized database, which is used to store profile of VSI type and VM. A VSI type is a set of policies or resource definition that can be shared by all VMs that choose to use this VSI type. VSI type can be regarded as an instance of Virtual Network. The profile is quite flexible, and it can be organized in a way shown in the following figure and include one or more of the following information. There can be other kind of profile organization format. The profile is very easy to extend to include more information.

VSI type	Profile type	description
VN1	Priority	The priority of traffic
	QoS	QoS policies for the VSI type
	ACL	ACL rules for the VSI type
	Bandwidth	Bandwidth of the traffic
	Multicast Addr	The multicast addr for all VMs belong to the VN
	VNID	A global unique ID for this VN
VN2	Priority	The priority of traffic
	QoS	QoS policies for the VSI type
	ACL	ACL rules for the VSI type
	Bandwidth	Bandwidth of the traffic
	Multicast Addr	The multicast addr for all VMs belong to the VSI type
	VNID	A global unique ID for this VN

Fig4. Profile organization example

A mapping between VSI type and VM is also managed on the database.

VSI type	VM list	Profile type	description
VN1	VM1	MAC Addr	The MAC Addr of VM's vNIC.
		VID	The VID to which the VM is associated.
		Inner Addr	The inner addr of the VM, which can be IPv4/v6 addr.
		Outer Addr	The outer addr of the VM, which can be IPv4/v6 addr.
	VM2	MAC Addr	The MAC Addr of VM's vNIC.
		VID	The VID to which the VM is associated.
		Inner Addr	The inner addr of the VM, which can be IPv4/v6 addr.
		Outer Addr	The outer addr of the VM, which can be IPv4/v6 addr.

Fig5. VSI type to VM mapping

The work flow of VDP with profile database is as follows.



```

TES/VM      NVE      database
|--start up-->|
or immigrate |<-get mappings and policies->|
              (VNID, inner to outer, etc)
              locally create
              caches
              |--register NVE-VNI mapping----->|
                                                Controller
                                                locally update
                                                NVE-VNI mapping

|-data frame->|
              |--encapsulation----->

|-emigrate--->|
              |--notify VM emigration----->|
              locally update                  locally update
              caches                          NVE-VNI mapping

                                                |-syn->|
                                                while mappings and/or
                                                policies is updated

              |<-synch mappings and policies-----|

              |<-get mappings and policies->|
              (VNID, inner to outer, etc)
              locally update
              caches

```

Fig7. Example work flow

## 5. Summary

Compared the mechanism in Sec 4.1 and 4.2, we can get the following results. From architecture view, exclusive controller has simpler architecture with few interaction requirements, and simpler work flow.

From performance view and reusing of existed protocols, hybrid mechanism is able to offload the query of static information to database, which can optimize the performance of controller and make the system more extensible.

## 6. Security Considerations

TBA

## 7. References

### 7.1. Normative Reference

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

[Qbg] "IEEE P802.1Qbg Edge Virtual Bridging".

### 7.2. Informative Reference

[framework]

Marc Lasserre, Marc., Balus, Florin., Morin, Thomas., Bitar, Nabil., and Yakov. Rekhter, "draft-lasserre-nvo3-framework-02", June 2012.

[overlay-cp]

Kreeger, L., Dutt, D., Narten, T., Black, D., and M. Sridharan, "draft-kreeger-nvo3-overlay-cp-00", Jan 2012.

[tes-nve-mechanism]

Gu, Y., "The mechanism and protocol between TES and NVE to facilitate NVO3", July 2012.

## Authors' Addresses

Gu Yingjie  
Huawei  
No. 101 Software Avenue  
Nanjing, Jiangsu Province 210001  
P.R.China

Phone: +86-25-56625392  
Email: guyingjie@huawei.com

Weiguo Hao  
Huawei



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 22, 2013

Y. Gu  
Y. Li  
Huawei  
Oct 19, 2012

The mechanism and signalling between TES and NVE  
draft-gu-nvo3-tes-nve-mechanism-01

Abstract

This draft introduces the interaction required between TES to NVE when NVE is located in an external box to TES. The signaling between TES and NVE has to be designed carefully to reflect all the interaction requirements. This document describes the relevant considerations for such design and also provides a basic analysis of the potential reusable protocols. Currently this draft focuses on the general interaction procedures with relevant parameters and the signaling design consideration. It may be extended to show more detailed signalling design recommendation and/or solution recommendation in the future with the progress of NVO3's work.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 22, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminologies and concepts . . . . .	6
3. TES to NVE Interaction . . . . .	9
3.1. Interaction Intentions . . . . .	9
3.2. VM Lifetime Events . . . . .	9
3.2.1. VM Creation . . . . .	9
3.2.2. VM Pre-associate with NVE . . . . .	10
3.2.3. VM Associate with NVE . . . . .	10
3.2.4. VM Suspension . . . . .	10
3.2.5. VM Resume . . . . .	11
3.2.6. VM Migration . . . . .	11
3.2.7. VM Termination . . . . .	11
3.2.8. VM Full Lifecycle Sketch . . . . .	11
3.3. Events, Interaction and Parameters . . . . .	13
3.3.1. VM Pre-association . . . . .	13
3.3.2. VM Association . . . . .	14
3.3.3. VM Suspension . . . . .	15
3.3.4. VM Resume . . . . .	15
3.3.5. VM Emigration . . . . .	16
3.3.6. VM Immigration . . . . .	16
3.3.7. VM Termination . . . . .	17
3.3.8. Keep-alive . . . . .	17
3.3.9. NVE Local Changes . . . . .	18
3.4. Signalling Design Considerations . . . . .	18
3.4.1. General Requirements . . . . .	18
3.4.2. Consideration . . . . .	19
3.4.3. Signalling States Machine . . . . .	19
4. Security Considerations . . . . .	20
5. Appendix 1: Mechanism Analysis . . . . .	20
5.1. IEEE 802.1Qbg . . . . .	20
5.1.1. Brief Introduction . . . . .	21
5.2. BGP . . . . .	23
5.3. External Controller . . . . .	23
6. References . . . . .	23
6.1. Normative Reference . . . . .	23
6.2. Informative Reference . . . . .	23
Authors' Addresses . . . . .	24



## 1. Introduction

Tenant End System (TES) is the physical host where tenant deploys their applications. Tenants' applications can be deployed on a physical server directly or on a virtual machine resided on a physical server. Tenant's virtual network, or say virtual data center, is an overlay network which is built on the underlying network, but logically independent of the underlying network. Network Virtualization Edge (NVE) is implemented with virtualization functions to encapsulate or decapsulate a tenant's packet that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses). A Tenant End System attaches to a Network Virtualization Edge (NVE) node, either directly or via a switched network (typically Ethernet). TES and NVE can be on the same physical server or on the separate devices. Fig1 to Fig3 show different NVE location cases. While TES and NVE are on the same physical server, the interaction between TES and NVE is via some proprietary internal interface which does not require a standard signaling protocol. Therefore such scenario is not the target of this document. For all the other scenarios, as long as the signaling between TES and NVE is visible to network developer, it is in the scope of this draft. We tried to examine the different locations of NVE to make sure the signaling interaction between NVE and TES cover as possible scenarios as possible.

- o (NVE Location 1) NVE and TES are co-located in a physical server. VM connects to NVE on Hypervisor. In this case, there should be some mechanism to assist Hypervisor know of VM changes, including adding, deleting and migration. Both VM and Hypervisor, as well as network service appliance, are controlled by VM Manager. VM Manager is aware of any VM identity and event, hence it can easily notify NVE about the information through some internal interface. A publically available standard protocol is not necessary in this case. Refer to Fig1.

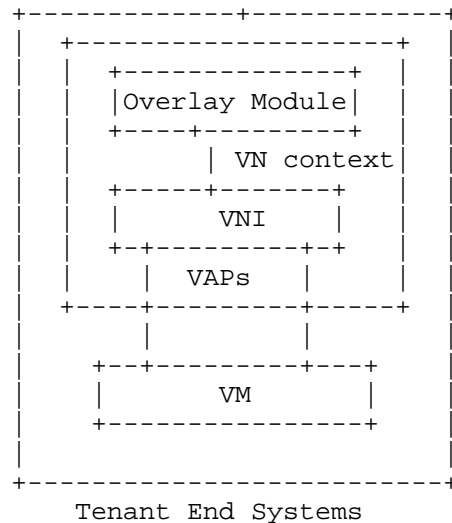


Figure 1

- o (NVE Location 2) TES connects to NVE on an external network entity next to it. VM is controlled by VM Manager, while NVE is controlled by some other management entity like network management system. Hence proprietary protocol between TES and NVE may not fit all the scenarios. A standard protocol to signal between TES and NVE is mandatory in this case. Refer to Fig2.

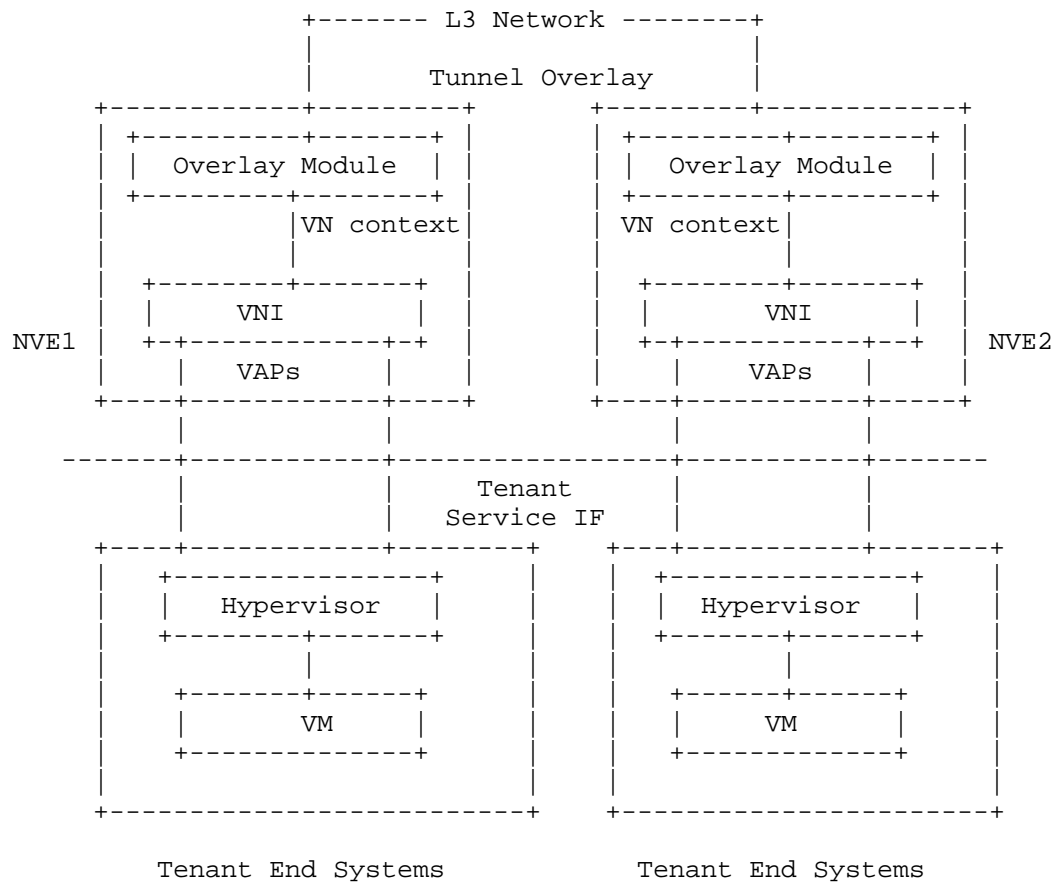


Figure 2: NVE Location3: VM connects to NVE on external network entity

- o (NVE Location 3) TES and NVE are indirectly connected. Refer to Fig3.

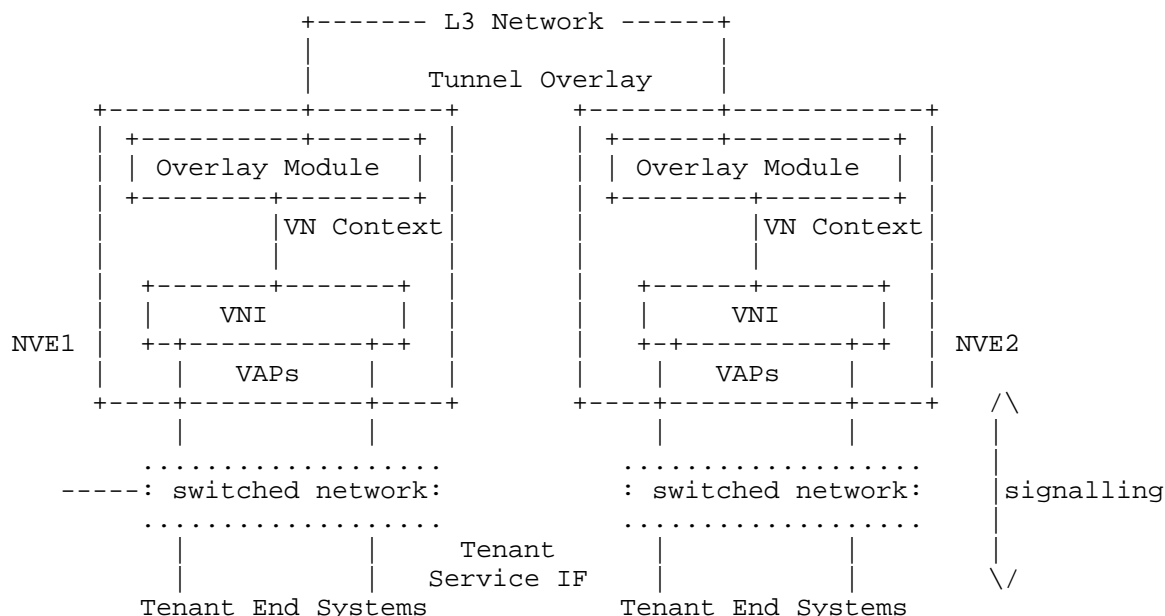


Figure 3: Reference model when TES and NVE are indirectly connected

In the mail list discussion, more than one mechanisms to be used between TES and NVE were discussed, including VDP (VSI Discovery and Configuration Protocol), BGP and others.. This draft is not going to make assertion about which protocol is better. We believe that each candidate protocol can, with some revision or updating, be used to exchange necessary events and information between TES and NVE. The final decision on which one to be used does not only depend on functionalities, but also some other aspects, e.g. lightweight to be implemented on server, widely deployment in the industry, efficiency and performance etc.

This draft first presents the recommended procedures of the TES and NVE signalling, key parameters of each step, and issues need to be addressed. Then a set of signaling design considerations are provided, which can be used as design requirements for the future signalling definition. In the appendix, we give a brief analysis on two existing protocols and also show how they can be revised to adapt to TES and NVE signaling.

## 2. Terminologies and concepts

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The document uses terms defined in [framework].

**VN:** Virtual Network. This is a virtual L2 or L3 domain that belongs a tenant.

**VNI:** Virtual Network Instance. This is one instance of a virtual overlay network. Two Virtual Networks are isolated from one another and may use overlapping addresses.

**Virtual Network Context or VN Context:** Field that is part of the overlay encapsulation header which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE. The egress NVE uses this field to determine the appropriate virtual network context in which to process the packet. This field MAY be an explicit, unique (to the administrative domain) virtual network identifier (VNID) or MAY express the necessary context information in other ways (e.g. a locally significant identifier).

**VNID:** Virtual Network Identifier. In the case where the VN context has global significance, this is the ID value that is carried in each data packet in the overlay encapsulation that identifies the Virtual Network the packet belongs to.

**NVE:** Network Virtualization Edge. It is a network entity that sits on the edge of the NVO3 network. It implements network virtualization functions that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses). An NVE could be implemented as part of a virtual switch within a hypervisor, a physical switch or router, a Network Service Appliance or even be embedded within an End Station.

**Underlay or Underlying Network:** This is the network that provides the connectivity between NVEs. The Underlying Network can be completely unaware of the overlay packets. Addresses within the Underlying Network are also referred to as "outer addresses" because they exist in the outer encapsulation. The Underlying Network can use a completely different protocol (and address family) from that of the overlay.

**Data Center (DC):** A physical complex housing physical servers, network switches and routers, Network Service Appliances and networked storage. The purpose of a Data Center is to provide application and/or compute and/or storage services. One such service is virtualized data center services, also known as Infrastructure as

a Service.

VM: Virtual Machine. Several Virtual Machines can share the resources of a single physical computer server using the services of a Hypervisor (see below definition).

Hypervisor: Server virtualization software running on a physical compute server that hosts Virtual Machines. The hypervisor provides shared compute/memory/storage and network connectivity to the VMs that it hosts. Hypervisors often embed a Virtual Switch (see below).

Virtual Switch: A function within a Hypervisor (typically implemented in software) that provides similar services to a physical Ethernet switch. It switches Ethernet frames between VMs' virtual NICs within the same physical server, or between a VM and a physical NIC card connecting the server to a physical Ethernet switch. It also enforces network isolation between VMs that should not communicate with each other.

Tenant: A customer who consumes virtualized data center services offered by a cloud service provider. A single tenant may consume one or more Virtual Data Centers hosted by the same cloud service provider.

Tenant End System: It defines an end system of a particular tenant, which can be for instance a virtual machine (VM), a non-virtualized server, or a physical appliance.

Virtual Access Points (VAPs): Tenant End Systems are connected to the Tenant Instance through Virtual Access Points (VAPs). The VAPs can be in reality physical ports on a ToR or virtual ports identified through logical interface identifiers (VLANs, internal VSwitch Interface ID leading to a VM).

VN Name: A globally unique name for a VN. The VN Name is not carried in data packets originating from End Stations, but must be mapped into an appropriate VN-ID for a particular encapsulating technology. Using VN Names rather than VN-IDs to identify VNs in configuration files and control protocols increases the portability of a VDC and its associated VNs when moving among different administrative domains (e.g. switching to a different cloud service provider).

VSI: Virtual Station Interface. Typically, a VSI is a virtual NIC connected directly with a VM. [Qbg]

### 3. TES to NVE Interaction

#### 3.1. Interaction Intentions

While TES is a non-virtualized physical server, a single physical interface on NVE is exclusively attached to a single tenant and the attachment doesn't change very frequently. In this case, NVE can be pre-configured with tenant's network properties and policies to execute appropriate packet processing. And when a physical server moves, which means a server change its attach point to the network, the new NVE, to which the server is going to attach with in the new location, can also be preconfigured. In this case, there is no need to proceed signalling between TES and NVE.

While TES is a virtualized server with multiple VMs, the interaction between TES and NVE becomes necessary. A physical interface on NVE can be attached to multiple VMs, which could belong to the same or different tenants, and VMs can be moved to new locations without physical shutdown, which means NVE not able to know VMs' attachment and/or detachment by checking the physical port. As described in [framework], NVE need to establish Virtual Network Instance for each tenant virtual network attached to it through physical interface, NVE must be able to know which tenants are attached to it and the corresponding VMs belongs to each tenants. So that NVE must be able to 1) identify and distinguish VMs attached to NVE through the same physical interface; 2) identify which tenant the VM belongs to; 3) get the network policies that is associated with the tenant. That's why a interaction signalling between TES and NVE is needed. Of course the signalling between TES and NVE are not limited to the above intentions. While looking into the detail processing of VM events, we will find more signalling functionalities and processing on TES and NVE.

#### 3.2. VM Lifetime Events

Not every VM has to pass through all the listed VM lifetime events. Any VM can have at least two or a combination of the following events.

##### 3.2.1. VM Creation

VM Manager indicates the hypervisor to schedule resources on server for a particular VM, including CPU, Memory, Storage and Network resources. After the VM is created on the server, the VM has necessary resource and is ready to be launched. The creation of VM doesn't necessarily mean the VM is running. The VM can be created but not launched for some while as long as the manager would like. The VM can be created and launched at once. Launching a VM just like

startup a physical computer.

Though VM creation is a very important events for VM, but the attached NVE needn't be aware of this event.

### 3.2.2. VM Pre-associate with NVE

VM Manager can decide when to launch a VM and connect the VM to the network. Before VM connects to network, operator need to provision VM's network properties and policies to the NVE that the VM is attached to. The examples of network properties are VM MAC address, tenant virtual network identifier. The examples of policies are ACL and QoS. But these properties and policies are not immediately activated on NVE unless the VM Manager indicate the VM to connect to network. This is called Pre-association. Pre-association is optional event.

### 3.2.3. VM Associate with NVE

This event means the VM is going to connect to the network. NVE has to get VM's network properties and policies, assign resources and install these properties and policies. If there is Pre-association before Association, NVE can reduce the time for Association. While VM is associated, it can use network resources as a physical server does.

Association can happen with or without pre-association. If there is Pre-association before Association, NVE has already the network properties and policies restored, or even installed. If the network properties and policies in Association message is the same as the pre-association, NVE can activate the installed network properties and policies. If they are different, the old reserved resources should be released and the new network properties and policies are installed and activated.

### 3.2.4. VM Suspension

Creating and terminating VM may take a considerable amount of time. Instead of performing these operations, operators can suspend a virtual machine for the required time and quickly resume it later. Suspending a VM is similar to putting a real computer into the sleep mode. When suspending a VM, VM's current state (including the state of all applications and processes running in the VM) is stored. When the suspended virtual machine is resumed, it continues operating at the same point the virtual machine was at the time of its suspending.



### 3.2.5. VM Resume

To activate the suspended VM. The suspended applications will start again at the state the VM was suspended. It's not always predictable on when a suspended VM will be resumed.

### 3.2.6. VM Migration

Two kinds VM migration, i.e. hot migration (or live migration) and offline migration. The processing of offline migration is similar to terminating the VM on one server and creating it on another server. The running applications on the VM will be broken and then be restarted again on the new location. For live migration, VM is lively migrated from one location to another, and the running applications should not be visibly disrupted. There is no termination or creation during live migration, so it's highly important to let NVE be aware of the migration so that corresponding network properties and policies can be correctly obtained, installed and activated on new location, and removed from the old location. Otherwise, there might be security risk and will influence or even interrupted running applications.

There are two sub-type for VM migration: VM emigration and VM immigration.

- o VM Emigrating: VM is emigrating from this server. Hence, all the relevant resources on the server and attached NVE are disabled, but not removed right now, and is ready to be removed once VM is successfully migrated. If VM is failed to immigrate on the new location, VM has to be resumed on old location with the states and policies disabled by old NVE.
- o VM Immigrating: VM is immigrating to this server. The server and attached NVE has prepared the necessary resources and is ready to enable the VM's properties and policies once VM is successfully migrated.

### 3.2.7. VM Termination

All applications and processing on VM is terminated. All VM's resources on server, including CPU, Memory, Storage and network resources, are released. There is no such a VM any more.

### 3.2.8. VM Full Lifecycle Sketch

Not every VM has to pass through all the lifetime events emulated in above. A simplest VM life has only VM Creation, VM Associating with NVE and VM Termination. A most complex VM life has all the events

listed in above. In this section, we show a sketch for a VM's full lifecycle with all listed events. This is helpful for the signalling designation in the future.

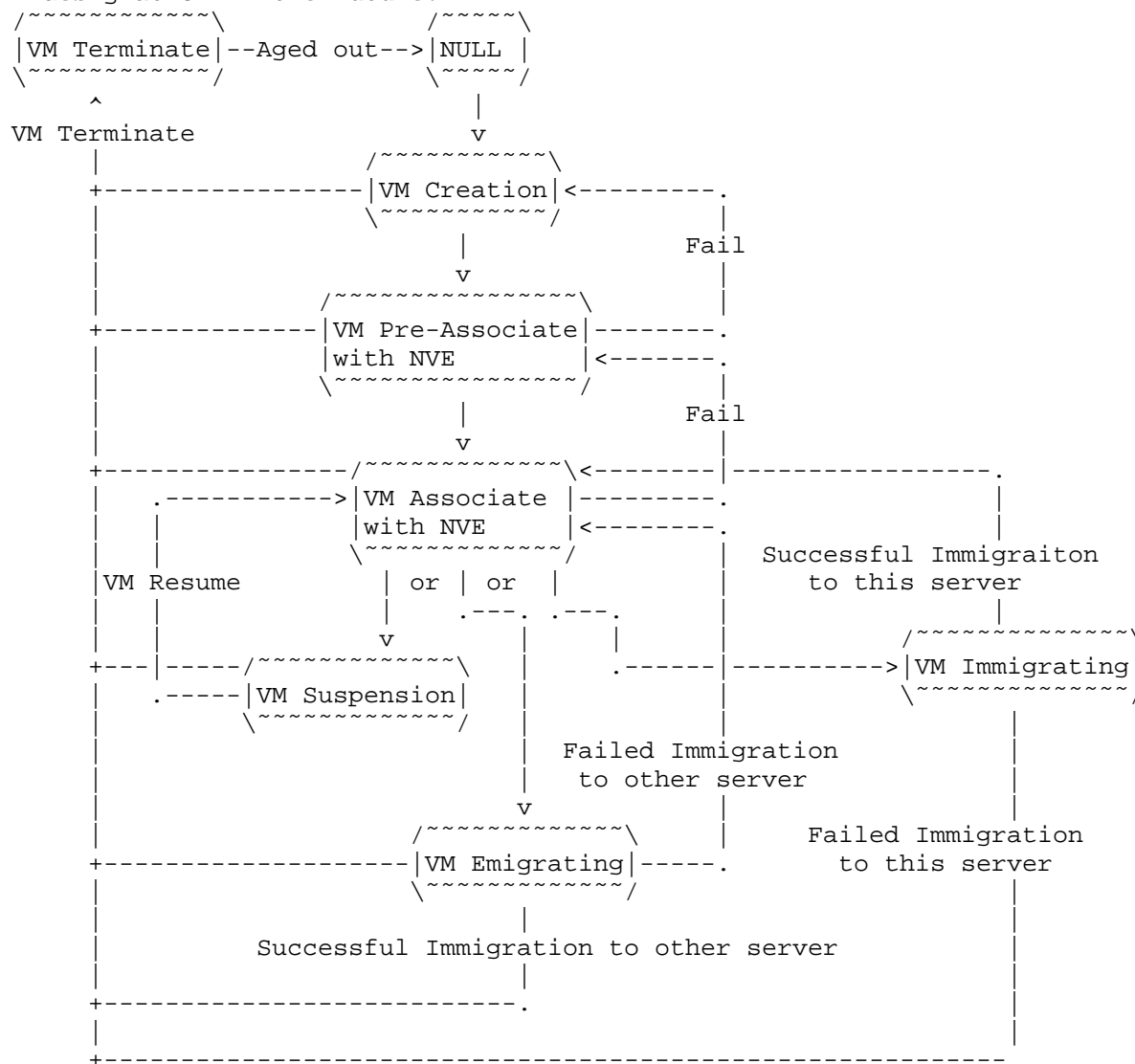


Figure 4: VM Full Lifecycle Sketch

### 3.3. Events, Interaction and Parameters

In this section, we will present description of interaction, parameters and special concerns for each VM events are provided. The interaction has strong relationship with VM lifetime events, but is not one-to-one mapping, for example, there is no interaction for VM Creation. For VM events, the interaction is initiated by hypervisor on behalf of a VM and sent to VNI on attached NVE. But this is not always the case, since NVE may also initiate interaction if there is some changes happen on NVE and those changes must be learned by particular VMs.

#### 3.3.1. VM Pre-association

- o Interaction: This event will trigger Hypervisor to compose a pre-association message, and then Hypervisor sends the message to NVE. While receives the pre-association message, NVE needs to authorize the VM and/or Hypervisor, obtain VM's network properties and policies, and install the properties and policies on NVE.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Pre-association.
  - \* VMID, a global unique ID in Data Center for a VM. A VM can have more than one MAC addresses and belongs to more than one VNID, so a VMID is necessary for NVE to accosicate the VNIDs and MACs with the particular VM.
  - \* VNID(s), a global unique ID in Data Center for a tenant's virtual network.
  - \* MAC addresses, a VM may have more than one MAC addresses. A VM may also belongs to more than one virtual network. So the MAC address(s) and VNID should be presented in a way that NVE can identify which MAC addresses belongs to which VNID.
  - \* Policies, including ACL, QoS, Priority and etc. In the case there are more than one VNID associated with the VM, Policies should be explicitly indicated to belong to which VNID.
- o Response: After NVE processes pre-association message, it repond to TES with processing result. The response can be SUCCESS or FAIL with such indicated reasons as FAILED AUTHORIZTION, CONFLICT POLICIES(e.g. the provisioned policies are conflict with other existed policies on NVE), NON-SUFFICIENT RESOURCES(e.g. the NVE has not enough resources to install the provisioned policies).

### 3.3.2. VM Association

- o Interaction: This event will trigger Hypervisor to compose an Association message, and then Hypervisor sends the message to NVE. Association can happen with or without a Pre-association message.
  - \* If there is a Pre-association message before Association, NVE needs to compare the information provided by Pre-association and Association. If they are same, NVE can activate the pre-installed resources. If they are different, NVE needs to do some additional work depending on what information has been changed from pre-association to association. For example, if policy or VNID is changed, NVE needs to update its memory.
  - \* If there is no Pre-association message before Association, NVE needs to do authorization, obtain VM's network properties and policies, and install and activate the properties and policies on NVE.
  - \* If there is another successful Association message before this Association, NVE needs to compare the information provided by previous provisioned Association and this Association. If all is the same, NVE do nothing except for update the VM's timer. If there is different in comparison, NVE needs to do some additional work, depends on what information is changed. For example, if policies or VNID is changed, NVE needs to update its memory.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Association.
  - \* VMID
  - \* VNID(s)
  - \* MAC addresses
  - \* Policies
- o Response: After NVE processes Association message, it repond to TES with processing result. The response can be SUCCESS or FAIL with such indicated reasons as FAILED AUTHORIZATION, CONFLICT POLICIES(e.g. the provisioned policies are conflict with other existed policies on NVE), NON-SUFFICIENT RESOURCES(e.g. the NVE has not enough resources to install the provisioned policies).

### 3.3.3. VM Suspension

- o Interaction: This event will trigger Hypervisor to compose an Suspension message or an Association message with Suspension indication, and then Hypervisor sends the message to NVE. Suspension must happen after Successful Association. On receiving a Suspension message, NVE inactivate, but not remove, the VM's resources and prepare for the next Resume message. In the state of suspension, NVE acts similar as it in Pre-association state. The FDB can be aged out during VM suspension.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Suspension or an Association message with Suspension indication
  - \* VMID
- o Response: After NVE processes Suspension message, it repond to TES with processing result. The response can be SUCCESS or FAIL . If it's FAIL, it may be because the NVE is too busy to process the message.

### 3.3.4. VM Resume

- o Interaction: This event will trigger Hypervisor to compose an Resume message or an Association message with Resume indication, and then Hypervisor sends the message to NVE. Resume is supposed to happen after a successful Suspension message, otherwise, it will be responded with a SUCCESS message and NVE will do nothing to the message.. On receiving a Resume message, NVE activates the VM's resources and prepare.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Resume or an Association message with Resume indication
  - \* VMID
- o Response: After NVE processes Resume message, it repond to TES with processing result. The response can be SUCCESS or FAIL. If it's FAIL, it may be because the NVE is too busy to process the message.

### 3.3.5. VM Emigration

- o Interaction: This event will trigger Hypervisor to compose an Emigration message or an Association message with Emigration indication, and then Hypervisor sends the message to NVE. Emigration can happen after Pre-association, Association, Suspension or Resume.
- o On receiving VM Emigration message or indication, NVE inactivate VM's resources. But NVE doesn't immediately remove VM's resources and states, because an emigration maybe fail if the immigration on the remote server or NVE is failed. In that case, the emigrating VM may need to continue its work on the current server. NVE will wait for a next Termination message to remove the VM's resources or states on NVE.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Association.
  - \* VMID
- o Response: After NVE processes VM Emigration, it repond to TES with processing result. The response can be SUCCESS or FAIL. If it's FAIL, it may be because the NVE is too busy to process the message.

### 3.3.6. VM Immigration

- o Interaction: This event will trigger Hypervisor to compose an Immigration message, or an Pre-association/Association message with Immigration indication, call them immigration(Pre-asso) and Immigration(Asso). NVE's reaction to VM Immigration is silimar to its reaction to Pre-association or Association. If the result of Immigration processing is FAIL, the VM will not migrate to the new location and continue its work on old server. VM Manger may have to find another new location for the VM to migrate to.
- o To distinguish Immigration from Pre-association and Association is meaningful, [statemigration-framework]shows the problem of VM's flow-coupled state migration in case of VM live migration. The Immigration message can be a indication or trigger for the flow-coupled state migration on middleboxes.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.

- \* Operation, i.e. Immigration or an (Pre-)Association message with Immigration indication.
- \* VMID
- \* VNID(s)
- \* MAC addresses
- \* Policies
- o Response: After NVE processes Immigration message, it repond to TES with processing result. The response can be SUCCESS or FAIL with such indicated reasons as FAILED AUTHORIZTION, CONFLICT POLICIES(e.g. the provisioned policies are conflict with other existed policies on NVE), NON-SUFFICIENT RESOURCES(e.g. the NVE has not enough resources to install the provisioned policies).

#### 3.3.7. VM Termination

- o Interaction: This event will trigger Hypervisor to compose an Termination message. NVE' will release VM's resources on NVE and remove all state about this VM.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Termination
  - \* VMID
- o Response: After NVE processes Termination message, it repond to TES with processing result. The response can be SUCCESS or FAIL. If it's FAIL, it maybe because NVE is too busy to process the Termination message, however the VM can be terminated on the server anyway.

#### 3.3.8. Keep-alive

This is not a VM lifetime events. Since the resources on NVE is precious, if a associated, pre-associated or suspended VM keeps idle for a pre-defined time, NVE will remove the VM's resources, so that NVE can serve other active VMs. In order to keep VM's resource on NVE, Hypervisor has to create keep-alive message, or an Pre-association/Association message with Keep-alive indication, NVE will update VM's timer upon the Keep-alive message.

Parameters: The signalling from TES to NVE should at least include

the following mandatory parameters.

- o Operation, i.e. Keep-alive or an (Pre-)Association message with Keep-alive indication.
- o VMID

### 3.3.9. NVE Local Changes

While VM associate with a VNID on NVE, NVE will generate local significant indicators for the VM and VNIDs, e.g. VID. If the indicators are sent to Hypervisor in previous response, and the indicators change later on, NVE need to create an Associate or a dedicated message with the changed indicators and send to Hypervisor, and Hypervisor will respond with processing result.

Note: Although we use the VM Lifetime events names as the names of messages in this section, it does mean that there should be a dedicated message for each event in the future signalling. Some of the events can be carried in one signalled message with different operation type. For example, an Association message with Immigration indication or an Association message with Suspension indication.

## 3.4. Signalling Design Considerations

### 3.4.1. General Requirements

#### 3.4.1.1. Basic Requirements

REQUIREMENT-1: The TNS (TES to NVE Signalling) MUST support TES to notify NVE about the VM's events, including but not limited to Pre-Association, Association, Emigration, Immigration and Termination.

REQUIREMENT-2: The TNS MUST support TES to notify NVE about the VM's VNID, which can be one identifier or a combination of several identifier.

REQUIREMENT-3: The TNS MUST support TES to notify NVE about the VM's address. The address MUST include one or both of MAC address of VM's virtual NIC and VM's IP address. And it SHOULD be extensible to carry new address type.

REQUIREMENT-4: The TNS MUST support NVE to notify TES about the VM's local tag. The local Tag type supported by TNP MUST include IEEE 802.1Q tag. And it SHOULD be extensible to carry other type of local tag.



#### 3.4.1.2. Extension Requirements

REQUIREMENT-5: The TNS SHOULD support NVE to notify TES about the VM's traffic PCP value.

In typical DC, where physical server connects to adjacent bridge, the data frame from server can be tagged with PCP or untagged. If a data frame is untagged, it can be tagged with PCP on adjacent bridge. While in virtualized DC, the adjacent bridge is Hypervisor. There are two options to deal with PCP tag, 1) data frame is tagged with PCP by VM, 2) data frame is tagged with PCP by Hypervisor and 3) data frame is tagged with PCP by NVE.

In cloud service, the VM can be anybody and it may want a higher priority than it should have. The VM can tag its data frame with higher PCP value and get better service. Based on the assumption that PCP provided by VM is not reliable, it's more reasonable to let the network to define the PCP value based on VM's priority, and enable bridges to tag the PCP value, as 2) or 3).

This problem is similar to local VID, which can be tagged either by Hypervisor or by NVE. The benefit to tag PCP by Hypervisor is to reduce the load on NVE.

#### 3.4.2. Consideration

To be added.

#### 3.4.3. Signalling States Machine

The interaction should be stateful. Both Hypervisor and NVE need to record the state of their signalling state. The main states are Pre-association, Association, Suspension, and Termination. The following diagram shows a the state machine of TES to NVE signalling. Only reasonable situations are listed in the diagram. In the future, more situation will be added to the state machine.

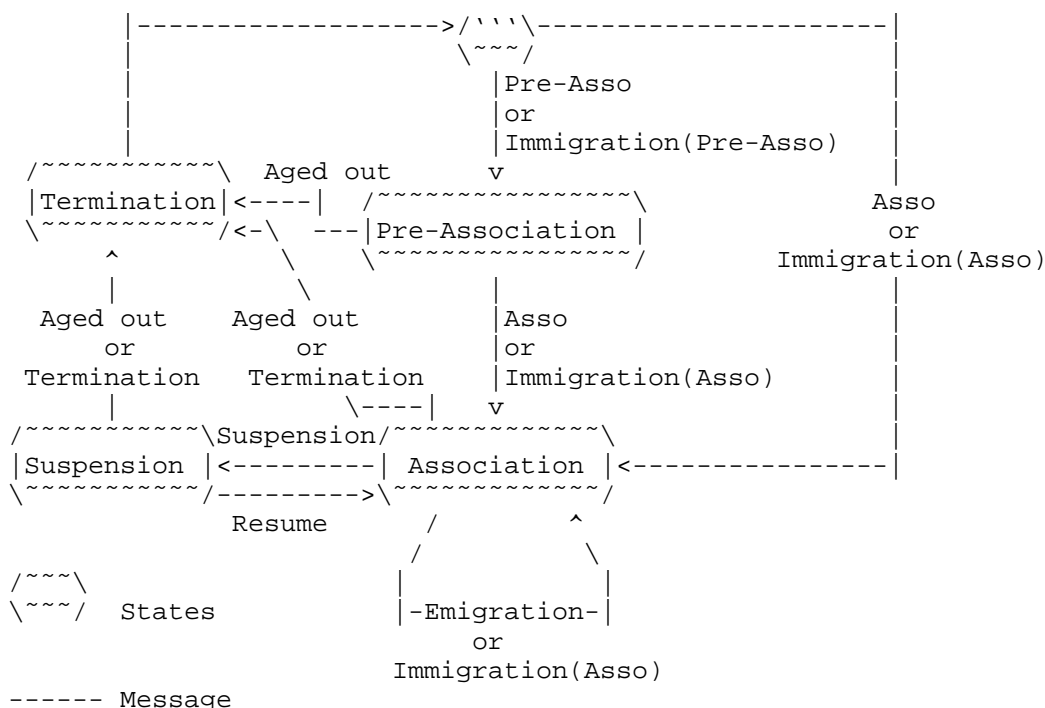


Figure 5: TES to NVE signalling State Machine

#### 4. Security Considerations

There are some considerations on security in [overlay-cp]. Most of the considerations are about mechanism between NVE and external controller, and the attack on underlying networks, which can not be resolved only by the mechanism between TES and NVE. One security issue related to the mechanism between TES and NVE is about the authentication of VM who announces to associate with a particular VN. There is a hypervisor between VMs and NVEs, and both VMs and hypervisor are not always reliable. For example, a poisoned hypervisor may modify the VN Name, or identification for similar intention, in order to associate with a VN that it doesn't belong to.

#### 5. Appendix 1: Mechanism Analysis

##### 5.1. IEEE 802.1Qbg

## 5.1.1.1. Brief Introduction

VDP has four basic TLV types.

- o Pre-Associate: Pre-Associate is used to pre-associate a VSI instance with a bridge port. The bridge validates the request and returns a failure Status in case of errors. Successful pre-association does not imply that the indicated VSI Type will be applied to any traffic flowing through the VSI. The pre-associate enables faster response to an associate, by allowing the bridge to obtain the VSI Type prior to an association.
- o Pre-Associate with resource reservation: Pre-Associate with Resource Reservation involves the same steps as Pre-Associate, but on successful pre-association also reserves resources in the Bridge to prepare for a subsequent Associate request.
- o Associate: The Associate TLV Type creates and activates an association between a VSI instance and a bridge port. The Bridge allocates any required bridge resources for the referenced VSI. The Bridge activates the configuration for the VSI Type ID. This association is then applied to the traffic flow to/from the VSI instance.
- o Deassociate: The de-associate TLV Type is used to remove an association between a VSI instance and a bridge port. Pre-Associated and Associated VSIs can be de-associated. De-associate releases any resources that were reserved as a result of prior Associate or Pre-Associate operations for that VSI instance.

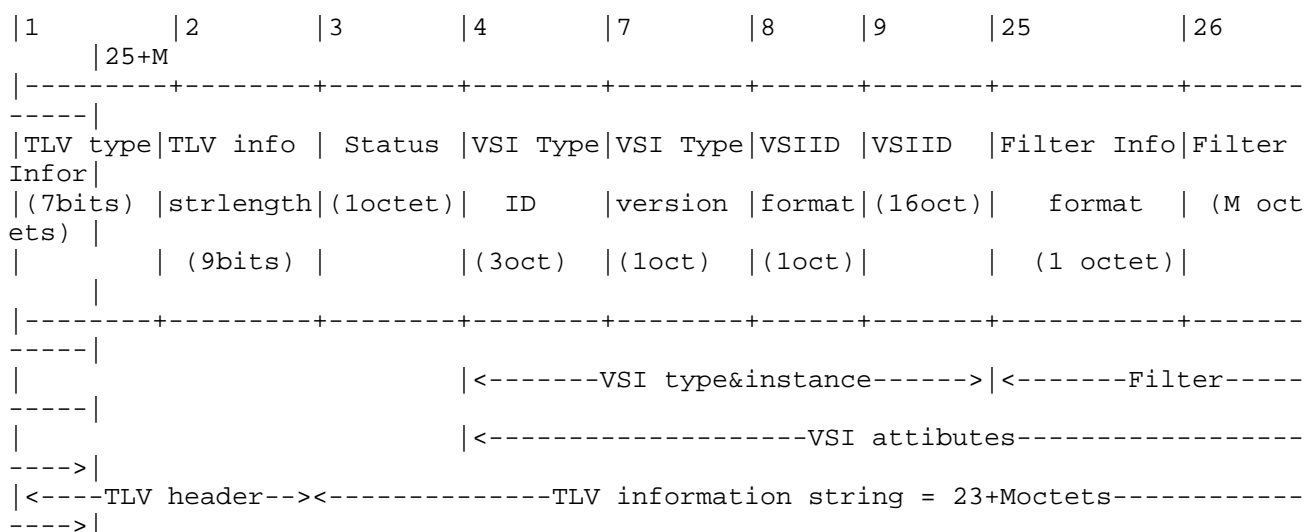


Figure 6: VDP TLV definitions

Some important flag values in VDP request:

- o M-bit (Bit 5): Indicates that the user of the VSI (e.g., the VM) is migrating (M-bit = 1) or provides no guidance on the migration of the user of the VSI (M-bit = 0). The M-bit is used as an indicator relative to the VSI that the user is migrating to.



- o S-bit (Bit 6): Indicates that the VSI user (e.g., the VM) is suspended (S-bit = 1) or provides no guidance as to whether the user of the VSI is suspended (S-bit = 0). A keep-alive Associate request with S-bit = 1 can be sent when the VSI user is suspended. The S-bit is used as an indicator relative to the VSI that the user is migrating from.

The filter information field supports the following format:

- o VID

#of entries (2octets)	PS (1bit)	PCP (3bits)	VID (12bits)
<--Repeated per entry-->			

Figure 7

- o MAC/VID

#of entries (2octets)	MAC address (6 octets)	PS (1bit)	PCP (3bits)	VID (12bits)
<-----Repeated per entry----->				

Figure 8

- o GroupID/VID

#of entries (2octets)	GroupID (4 octets)	PS (1bit)	PCP (3bits)	VID (12bits)
<-----Repeated per entry----->				

Figure 9

- o GroupID/MAC/VID

#of entries (2octets)	GroupID (4 octets)	MAC address (6 octets)	PS (1bit)	PCP (3bits)	VID (12bits)
<-----Repeated per entry----->					

Figure 10

In each format, the null VID can be used in the VDP Request. In this case, the Bridge is expected to supply the corresponding local VID value in the VDP Response.

The VSIID in VDP request that identify a VM can be one of the following format: IPV4 address, IPV6 address, MAC address, UUID or locally defined.

VDP features	Requirements Matching
Pre-Associate/ Pre-Associate with resource reservation/ Associate/ Deassociate	Requirement-1
M-bit/S-bit	Requirement-1
VSI type&instance in VDP request	Requirement-2
Filter Infor	Requirement-3
VID infor in VDP response	Requirement-4
PCP in VDP response	Requirement-5

#### VDP TLV types

### 5.2. BGP

gives a brief analysis on how BGP can be reused for TES and NVE signalling. Please refer to it for more information. [server2nve]

### 5.3. External Controller

## 6. References

### 6.1. Normative Reference

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

[Qbg] "IEEE P802.1Qbg Edge Virtual Bridging".

### 6.2. Informative Reference

[framework]

Marc Lasserre, Marc., Balus, Florin., Morin, Thomas., Bitar, Nabil., and Yakov. Rekhter, "draft-ietf-nvo3-framework-00", September 2012.

[overlay-cp]

Kreeger, L., Dutt, D., Narten, T., Black, D., and M.  
Sridharan, "draft-kreeger-nvo3-overlay-cp-00", Jan 2012.

[server2nve]

Kompella, K.,  
"draft-dunbar-nvo3-overlay-mobility-issues-00", July 2012.

[statemigration-framework]

Gu, Y., Shore, M., and S. Sivakumar, "A Framework and  
Problem Statement for Flow-associated Middlebox State  
Migration", October 2012.

#### Authors' Addresses

Gu Yingjie  
Huawei  
No. 101 Software Avenue  
Nanjing, Jiangsu Province 210001  
P.R.China

Phone: +86-25-56625392  
Email: guyingjie@huawei.com

Yizhou Li  
Huawei  
No. 101 Software Avenue  
Nanjing, Jiangsu Province 210001  
P.R.China

Phone:  
Email: liyizhou@huawei.com





Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: October 31, 2013

K. Kompella  
Y. Rekhter  
Juniper Networks  
T. Morin  
France Telecom - Orange Labs  
D. Black  
EMC Corporation  
April 29, 2013

Signaling Virtual Machine Activity to the Network Virtualization Edge  
draft-kompella-nvo3-server2nve-02

Abstract

This document proposes a simplified approach for provisioning network parameters related to Virtual Machine creation, migration and termination on servers. The idea is to provision the server, then have the server signal the requisite parameters to the relevant network device(s). Such an approach reduces the workload on the provisioning system and simplifies the data model that the provisioning system needs to maintain. It is also more resilient to topology changes in server-network connectivity, for example, reconnecting a server to a different network port or switch.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 31, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. VM Creation . . . . .	3
1.2. VM Live Migration . . . . .	4
1.3. VM Termination . . . . .	5
2. Acronyms Used . . . . .	6
3. Virtual Networks . . . . .	7
3.1. Current Mode of Operation . . . . .	8
3.2. Future Mode of Operation . . . . .	8
4. Provisioning DCVPNs . . . . .	9
5. Signaling . . . . .	9
5.1. Preliminaries . . . . .	9
5.2. VM Operations . . . . .	10
5.2.1. Network Parameters . . . . .	10
5.2.2. Creating a VM . . . . .	12
5.2.3. Terminating a VM . . . . .	14
5.2.4. Migrating a VM . . . . .	15
5.3. Signaling Protocols . . . . .	16
6. Interfacing with DCVPN Control Planes . . . . .	16
7. Security Considerations . . . . .	16
8. IANA Considerations . . . . .	17
9. Acknowledgments . . . . .	17
10. Informative References . . . . .	17
Authors' Addresses . . . . .	18

## 1. Introduction

To create a Virtual Machine (VM) on a server in a data center, one must specify parameters for the compute, storage, network and appliance aspects of the VM. At a minimum, this requires provisioning the server that will host the VM, and the Network Virtualization Edge (NVE) that will implement the virtual network for the VM in addition to the VM's storage. Similar considerations apply to live migration and terminating VMs. This document proposes mechanisms whereby a server can be provisioned with all of the parameters for the VM, and the server in turn signals the networking aspects to the NVE. The NVE may be located on the server or in an

external network switch that may be directly connected to the server or accessed via an L2 (Ethernet) LAN or VLAN. The following sections capture the abstract sequence of steps for VM creation, live migration and deletion.

While much of the material in this draft may apply to virtual entities other than virtual machines that exist on physical entities other than servers, this draft is written in terms of virtual machines and servers for clarity.

### 1.1. VM Creation

This section describes an abstract sequence of steps involved in creating a VM and making it operational (the latter is also known as "powering on" the VM). The following steps are intended as an illustrative example, not as prescriptive text; the goal is to capture sufficient detail to set a context for the signaling described in Section 5.

Creating a VM requires:

1. gathering the compute, network, storage, and appliance parameters required for the VM;
2. deciding which server, network, storage and network appliance devices best match the VM requirements in the current state of the data center;
3. provisioning the server with the VM parameters;
4. provisioning the network element(s) to which the server is connected with the network-related parameters of the VM;
5. informing the network element(s) to which the server is connected about the VM's peer VMs, storage devices and other network appliances with which the VM needs to communicate;
6. informing the network element(s) to which a VM's peer VMs are connected about the new VM and its addresses;
7. provisioning storage with the storage-related parameters; and
8. provisioning necessary network appliances (firewalls, load balancers and "middle boxes").

Steps 1 and 2 are primarily information gathering. For Steps 3 to 8, the provisioning system talks actively to servers, network switches, storage and appliances, and must know the details of the physical

server, network, storage and appliance connectivity topologies. Step 4 is typically done using just provisioning, whereas Steps 5 and 6 may be a combination of provisioning and other techniques that may defer discovery of the relevant information. Steps 4 to 6 accomplish the task of provisioning the network for a VM, the result of which is a Data Center Virtual Private Network (DCVPN) overlaid on the physical network.

While shown as a numbered sequence above, some of these steps may be concurrent (e.g., server, storage and network provisioning for the new VM may be done concurrently), and the two "informing" steps for the network (5 and 6) may be partially or fully lazily evaluated based on network traffic that the VM sends or receives after it becomes operational.

This document focuses on the case where the network elements in Step 4 are not co-resident with the server, and describes how the provisioning in Step 4 can be replaced by signaling between server and network, using information from Step 3.

## 1.2. VM Live Migration

This subsection describes an abstract sequence of steps involved in live migration of a VM. Live migration is sometimes referred to as "hot" migration, in that from an external viewpoint, the VM appears to continue to run while being migrated to another server (e.g., TCP connections generally survive this class of migration). In contrast, suspend/resume (or "cold") migration consists of suspending VM execution on one server and resuming it on another. The following live migration steps are intended as an illustrative example, not as prescriptive text; the goal is to capture sufficient detail to provide context for the signaling described in Section 5.

For simplicity, this set of abstract steps assumes shared storage, so that the VM's storage is accessible to the source and destination servers. Live migration of a VM requires:

1. deciding which server should be the destination of the migration based on the VM's requirements, data center state and reason for the migration;
2. provisioning the destination server with the VM parameters and creating a VM to receive the live migration;
3. provisioning the network element(s) to which the destination server is connected with the network-related parameters of the VM;

4. transferring the VM's memory image between the source and destination servers;
5. actually moving the VM: pausing the VM's execution on the source server, transferring the VM's execution state and any remaining memory state to the destination server and continuing the VM's execution on the destination server;
6. informing the network element(s) to which the destination server is connected about the VM's peer VMs, storage devices and other network appliances with which the VM needs to communicate;
7. informing the network element(s) to which a VM's peer VMs are connected about the VM's new location;
8. activating the VM's network parameters at the destination server;
9. deprovisioning the VM from the network element(s) to which the source server is connected; and
10. deleting the VM from the source server.

Step 1 is primarily information gathering. For Steps 2, 3, 9 and 10, the provisioning system talks actively to servers, network switches and appliances, and must know the details of the physical server, network and appliance connectivity topologies. Steps 4 and 5 are usually handled directly by the servers involved. Steps 6 to 9 may be handled by the servers (e.g., one or more "gratuitous" ARPs or RARPs from the destination server may accomplish all four steps) or other techniques. For steps 6 and 7, the other techniques may involve discovery of the relevant information after the VM has been migrated.

While shown as a numbered sequence above, some of these steps may be concurrent (e.g., moving the VM and associated network changes), and the two "informing" steps (6 and 7) may be partially or fully lazily evaluated based on network traffic that the VM sends and/or receives after it is migrated to the destination server.

This document focuses on the case where the network elements are not co-resident with the server, and shows how the provisioning in Step 3 and the deprovisioning in Step 9 can be replaced by signaling between server and network, using information from Step 3.

### 1.3. VM Termination

This subsection describes an abstract sequence of steps involved in termination of a VM, also referred to as "powering off" a VM. The following termination steps are intended as an illustrative example, not as prescriptive text; the goal is to capture sufficient detail to set a context for the signaling described in Section 5.

Termination of a VM requires:

1. ensuring that the VM is no longer executing;
2. deprovisioning the VM from the network element(s) to which the server is connected; and
3. deleting the VM from the server (the VM's image may remain on storage for reuse).

Steps 1 and 3 are handled by the server, based on instructions from the provisioning system. For Step 2, the provisioning system talks actively to servers, network switches, storage and appliances, and must know the details of the physical server, network, storage and appliance connectivity topologies.

While shown as a numbered sequence above, some of these steps may be concurrent (e.g., network deprovisioning and VM deletion).

This document focuses on the case where the network elements in Step 2 are not co-resident with the server, and shows how the deprovisioning in Step 3 can be replaced by signaling between server and network.

## 2. Acronyms Used

The following acronyms are used:

DCVPN: Data Center Virtual Private Network -- a virtual connectivity topology overlaid on physical devices to provide virtual devices with the connectivity they need and isolation from other DCVPNs. This corresponds to the concept of a Virtual Network Instance (VNI) in [I-D.ietf-nvo3-framework].

NVE: Network Virtualization Edge -- the entities that realize private communication among VMs in a DCVPN

l-NVE: local NVE: wrt a VM, NVE elements to which it is directly connected

r-NVE: remote NVE: wrt a VM, NVE elements to which the VM's peer VMs are connected

NVGRE: Network Virtualization using Generic Routing Encapsulation

VDP: VSI Discovery and Configuration Protocol

VID: 12-bit VLAN tag or identifier used locally between a server and its l-NVE

VLAN: Virtual Local Area Network

VM: Virtual Machine (same as Virtual Station)

Peer VM: wrt a VM, other VMs in the VM's DCVPN

VNID: DCVPN Identifier

VSI: Virtual Station Interface

VXLAN: Virtual eXtensible Local Area Network

### 3. Virtual Networks

The goal of provisioning a network for VMs is to create an "isolation domain" wherein a group of VMs can talk freely to each other, but communication to and from VMs outside that group is restricted (either prohibited, or mediated via a router, firewall or other network gateway). Such an isolation domain, sometimes called a Closed User Group, here will be called a Data Center Virtual Private Network (DCVPN). The network elements on the outer border or edge of the overlay portion of a Virtual Network are called Network Virtualization Edges (NVEs).

A DCVPN is assigned a global "name" that identifies it in the management plane; this name is unique in the scope of the data center, but may be unique across several cooperating data centers. A DCVPN is also assigned an identifier unique in the scope of the data center, the Virtual Network Group ID (VNID). The VNID is a control plane entity. A data plane tag is also needed to distinguish different DCVPNs' traffic; more on this later.

For a given VM, the NVE can be classified into two parts: the network elements to which the VM's server is directly connected (the local NVE or l-NVE), and those to which peer VMs are connected (the remote NVE or r-NVE). In some cases, the l-NVE is co-resident with the server hosting the VM; in other cases, the l-NVE is separate (distributed l-NVE). The latter case is the one of primary interest in this document.

A created VM is added to a DCVPN through Steps 4 to 6 in section Section 1.1 which can be recast as follows. In Step 4, the l-NVE(s) are informed about the VM's VNID, network addresses and policies, and the l-NVE and server agree on how to distinguish traffic for different DCVPNs from and to the server. In Step 5 the relevant r-NVE elements and the addresses of their VMs are discovered, and in Step 6, the r-NVE(s) are informed of the presence of the new VM and obtain or discover its addresses; for both steps 5 and 6, the discovery may be lazily evaluated so that it occurs after the VM begins sending and receiving DCVPN traffic.

Once a DCVPN is created, the next steps for network provisioning are to create and apply policies such as for QoS or access control. These occur in three flavors: policies for all VMs in the group, policies for individual VMs, and policies for communication across DCVPN boundaries.

### 3.1. Current Mode of Operation

DCVPNs are often realized as Ethernet VLAN segments. A VLAN segment satisfies the communication properties of a DCVPN. A VLAN also has data plane mechanisms for discovering network elements (Layer 2 switches, aka bridges) and VM addresses. When a DCVPN is realized as a VLAN, Step 4 in section Section 1.1 requires provisioning both the server and l-NVE with the VLAN tag that identifies the DCVPN. Step 6 requires provisioning all involved network elements with the same VLAN tag. Address learning is done by flooding, and the announcement of a new VM or the new location of a migrated VM is often via a "gratuitous" ARP or RARP.

While VLANs are familiar and well-understood, they have scaling challenges because they are Layer 2 infrastructure. The number of independent VLANs in a Layer 2 domain is limited by the 12-bit size of the VLAN tag. In addition, data plane techniques (flooding and broadcast) are another source of scaling concerns as the overall size of the network grows.

### 3.2. Future Mode of Operation

There are multiple scalable realizations of DCVPNs that address the isolation requirements of DCVPNs as well as the need for a scalable substrate for DCVPNs and the need for scalable mechanisms for NVE and VM address discovery. While describing these approaches beyond the scope of this document, a secondary goal of this document is to show how the signaling that replaces Step 4 in section Section 1.1 can seamlessly interact with realizations of DCVPNs.



VLAN tags (VIDs) will be used as the data plane tag to distinguish traffic for different DCVPNs' between a server and its l-NVE. Note that, as used here, VIDs only have local significance between server and NVE, and should not be confused with data-center-wide usage of VLANs. If VLAN tags are used for traffic between NVEs, that tag usage depends on the encapsulation mechanism among the NVEs and is orthogonal to VLAN tag usage between servers and l-NVEs.

#### 4. Provisioning DCVPNs

For VM creation as described in section Section 1.1, Step 3 provisions the server; Steps 4 and 5 provision the l-NVE elements; Step 6 provisions the r-NVE elements.

In some cases, the l-NVE is located within the server (e.g., a software-implemented switch within a hypervisor); in this case, Steps 3 and 4 are "single-touch" in that the provisioning system need only talk to the server, as both compute and network parameters are applied by the server. However, in other cases, the l-NVE is separate from the server, requiring that the provisioning system talk to both the server and l-NVE. This scenario, which we call "distributed local NVE", is the one considered in this document. This draft's goal is to describe how "single-touch" provisioning can be achieved in the distributed l-NVE case.

The overall approach is to provision the server, and have the server signal the requisite parameters to the l-NVE. This approach reduces the workload on the provisioning system, allowing it to scale both in the number of elements it can manage, as well as the rate at which it can process changes. It also simplifies the data model of the network that is used by the provisioning system, because a complete, up-to-date map of server to network connectivity is not required. This approach is also more resilient to server-network connectivity/topology changes that have not yet been transmitted to the provisioning system. For example, if a server is reconnected to a different port or a different l-NVE to recover from a malfunctioning port, the server can contact the new l-NVE over the new port without the provisioning system needing to immediately be aware of the change.

While this draft focuses on provisioning networking parameters via signaling, extensions may address the provisioning of storage and network appliance parameters in a similar fashion.

#### 5. Signaling

##### 5.1. Preliminaries

This draft considers three common VM operations in a virtualized data center: creating a VM; migrating a VM from one physical server to another; and terminating a VM. Creating a VM requires "associating" it with its DCVPN and "activating" that association; decommissioning a VM requires "dissociating" the VM from its DCVPN. Moving a VM consists of associating it with its DCVPN in its new location, activating that association, and dissociating the VM from its old location.

## 5.2. VM Operations

### 5.2.1. Network Parameters

For each VM association or dissociation operation, a subset of the following information is needed from server to l-NVE:

operation: one of associate or dissociate.

authentication: proof that this operation was authorized by the provisioning system

VNID: identifier of DCVPN to which VM belongs

VID: tag to use between server and l-NVE to distinguish DCVPN traffic; the value zero in an associate operation is a request that the l-NVE to assign an unused VID. This approach provides extensibility by allowing the VID to be a VLAN-id, although other local means of multiplexing traffic between the server and the NVE could be used instead of VIDs.

encapsulation type: type of encapsulation used by the DCVPN for traffic exchanged between NVEs (see below).

network addresses: network addresses for VM on the server (e.g., MACs)

policy: VM-specific and/or network-address-specific network policies, such as access control lists and/or QoS policies

hold time: time (in milliseconds) to keep a VM's addresses after it migrates away from this l-NVE. This is usually set to zero when a VM is terminated.

per-address-VID-allocation: boolean flag which can optionally be set to "yes", resulting in the VID allocated to the this address being distinct from the VID allocated to other addresses (for the same VM or other VMs) connected to the same DCVPN on a same NVE port; this behavior will result in traffic always transiting through the

NVE, even to/from other addresses for the same DCVPN on the same server.

The "activate" operation is a dataplane operations that references a previously established association via the address and VID; all other parameters are obtained at the NVE by mapping the source address, VID and port involved to obtain information established by a prior associate operation.

Realizations of DCVPNs include, E-VPNs ([I-D.ietf-l2vpn-evpn]), IP VPNs ([RFC4364]), NVGRE ([I-D.sridharan-virtualization-nvgre], VPLS ([RFC4761], [RFC4762]), and VXLAN ([I-D.mahalingam-dutt-dcops-vxlan]). The encapsulation type determines whether forwarding at the NVE for the DCVPN is based on Layer 2 or Layer 3 service.

Typically, for the associate messages, all of the above information except hold time would be needed. Similarly, for the dissociate message, all of the above information except VID and encapsulation type would typically be needed.

These operations are stateful in that their results remain in place until superseded by another operation. For example, on receiving an associate message, an NVE is expected to create and maintain the DCVPN information for the addresses until the NVE receives a dissociate message to remove that information. A separate liveness protocol may be run between server and NVE to let each side know that the other is still operational; if the liveness protocol fails, each side may remove state installed in response to messages from the other.

The descriptions below generally assume that the NVEs participate in a mechanism for control plane distribution of VM addresses, as opposed to doing this in the data plane. If this is not the case, NVE elements can lazily evaluate (via data plane discovery) the parts of the procedures below that involve address distribution.

As VIDs are local to server-NVE communication, in fact to a specific port connecting these two elements, a mapping table containing 4-tuples of the following form will prove useful to the NVE:

<VID, port, VNID, VM network address>

The valid VID values are from 1 to 4094, inclusive. A value of 0 is used to mean "unassigned". When a VID can be shared by more than one VM, it is necessary to reference-count entries in this table; the list of addresses in an entry serves this purpose. Entries in this table have multiple uses:

- o Finding the VNID for a VID and port for association, activation and traffic forwarding;
- o Determining whether a VID exists (has already been assigned) for a VNID and port.
- o Determining which <VID, port> pairs to use for forwarding traffic that requires flooding on the DCVPN.

For simplicity and clarity, this draft assumes that the network interfaces in VMs (vNICs) do not use VLAN tags.

#### 5.2.2. Creating a VM

When a VM is instantiated on a server (powered on, e.g., after creation), each of the VM's interfaces is assigned a VNID, one or more network addresses and an encapsulation type for the DCVPN. The VM addresses may be any of IPv4, IPv6 and MAC addresses. There may also be network policies specific to the VM or its interfaces. To connect the VM to its DCVPN, the server signals these parameters to the l-NVE via an "associate" operation followed by an "activate" operation to put the parameters into use. (Note that the l-NVE may consist of more than one device.)

On receiving an associate message on port P from server S, an NVE device does the following for each network address in that message:

A.1: Validate the authentication (if present). If not, inform the provisioning system, log the error, and stop processing the associate message. This validation may include authorization checks.

A.2: Check the per-address-VID-allocation flag in the associate message:

- \* if this flag is not set:

- + Check if the VID in the associate message is zero (i.e., the associate message requests VID allocation); if so, look up the VID for <VNID, port, network address> ; if there is no current VID for that tuple, allocate a new VID

- + If the VID in the associate message is non-zero, look up the VID for <VNID, port>. If that lookup results in the same VID as the one in the associate message, associate that VID with <VNID, network address>. If the lookup indicates that there is no current VID for that tuple, associate the VID in the associate message with <VNID, port, network address>. Otherwise, the VID in the associate message does not match the VID that is currently in use for <VNID, port>, so respond to S with an error, and stop processing the associate message.
  - \* if this flag is set, check if the VID in the associate message is zero :
    - + if so, this is an allocation request, so allocate a new VID, distinct from other VIDs allocated on this port;
    - + if the VID is non-zero, check that the provided VID is distinct from other VIDs allocated on this port; if so, associate the VID with <VNID, port, network address>. If not, the provided VID is already in used and hence cannot be dedicated to this network address, so respond to S with an error, and stop processing the associate message.
- A.3: Add the <VID, port, VNID, network address> entry to the NVE's mapping table. This table entry includes information about the DCVPN encapsulation type for the VNID.
- A.4: Communicate with the control plane to advertise the network address, and (if the VNID is new to the NVE) also to get other network addresses in the DCVPN. Populate the NVE's mapping table with all of these network addresses (some control planes may not provide all or even any of the other addresses in the DCVPN at this point).
- A.5: Finally, respond to S with the VID for <VNID, port, network address>, and indicate that the operation was successful.

After a successful associate, the network has been provisioned (at least in the local NVE) for traffic, but forwarding has not been enabled. On receiving an activate message on port P from server S, an NVE device does the following (activate is a one-way message that does not have a response):

- B.1: Validate the authentication (if present). If not, inform the provisioning system, log the error, and stop processing the associate message. This validation may include authorization checks. The authentication and authorization may be implicit when

the activate message is a dataplane frame (e.g., a "gratuitous" ARP or RARP).

- B.2: Check if the VID in the activate message is zero. If so, log the error, and stop processing the activate message.
- B.3: Use the VID and port P to look up the VNID from a previous associate message. If there is no mapping table state for that VID and port, log the error and stop processing the activate message.
- B.4: If forwarding is not enabled for <VID, port, network address> activate it, mapping VID -> VNID on this port (P) for traffic sent to and received from r-NVEs.
- B.5: If the activate message is a dataplane frame that requires forwarding beyond the NVE, (e.g., a "gratuitous" ARP or RARP), use the activated forwarding to send the frame onward via the virtual network identified by the VNID.

#### 5.2.3. Terminating a VM

On receiving a request from the provisioning system to terminate execution of a VM (powering off the VM, whether or not the VM's image is retained on storage), the server sends a dissociate message to the l-NVE with the hold time set to zero. The dissociate message contains the operation, authentication, VNID, encapsulation type, and VM addresses. On receiving the dissociate message on port P from server S, each NVE device L does the following:

- D.1: Validate the authentication (if present). If not, inform the provisioning system, log the error, and stop processing the associate message.
- D.2: Communicate with the control plane to withdraw the VM's addresses. If the hold time is as non-zero, wait until the hold time expires before proceeding to the next step.
- D.3: Delete the VM's addresses from the mapping table and delete any VM-specific network policies associated with any of the VM addresses. If a mapping tuple contains no VM addresses as a result delete that tuple. If the mapping table contains no entries for the VNID involved after deleting the tuple, optionally delete any network policies for the VNID.
- D.4: Respond to S saying that the operation was successful.

At step D.2, the control plane is responsible for not disrupting network operation if the addresses are in use at another l-NVE. Also, l-NVEs cannot rely on receiving dissociate messages for all terminated VMs, as a server crash may implicitly terminate a VM before a dissociate message can be sent.

#### 5.2.4. Migrating a VM

Consider a VM that is being migrated from server S (connected to l-NVE device L) to server S' (connected to l-NVE device L'). This section assumes shared storage, so that both S and S' have access to the VM's storage. The sequence of steps for a successful VM migration is:

- M.1: S' gets a request to prepare to receive a copy of the VM from S.
- M.2: S gets a request to copy the VM to S'.
- M.3: The copy of the VM (memory, configuration state, etc.) occurs while the VM continues to execute.
- M.4: When that copy has made sufficient progress, S pauses the VM, and completes the copy, including the VM's execution state.
- M.5: S' gets a request to resume the paused VM.
- M.6: After that resume has succeeded, S then proceeds to terminate the paused VM on S, see section Section 5.2.3, but this operation may specify a non-zero hold time during which traffic received may be forwarded to the VM's new location.

Steps M.1 and M.2 initiate the copy of the VM. During step M.3, S' sends an "associate" message to L' for each of the VM's network addresses (S' receives information about these addresses as part of the VM copy). Step M.4 occurs when the VM copy has made sufficient progress that the pause required to transfer the VM's execution from S to S' is sufficiently short. At step M.4, or M.5 at the latest, S' sends an "activate" message to L' for each of the VM's interfaces. At Step M.6, S sends a "dissociate" message to L for each of the VM's network addresses, optionally with a non-zero hold time.

From the DCVPN's view, there are two important overlaps in the apparent network location of the VM's addresses:

- o The VM's addresses are associated with both L and L' between steps M.3 and M.6.

- o The VM's addresses are activated at L' during step M.4 or step M.5 at the latest (e.g., if activate is a dataplane operation based on traffic sent at that step); both of these typically occur before these addresses are dissociated at L during step M.6

The DCVPN control plane must work correctly in the presence of these overlaps, and in particular must not:

- o Fail to activate the VM's network addresses at L' because they have not yet been withdrawn at L, or
- o Disruptively withdraw the VM's network addresses from use at step M.6 of a migration when the VM continues to execute on a different server.

An additional scenario that is important for migration is that the source and destination servers, S and S', may share a common l-NVE, i.e., L and L' are the same. In this scenario there is no need for remote interaction of that l-NVE with other NVEs, but that NVE must be aware of the possibility of a new association of the VM's addresses with a different port and the need to promptly activate them on that port even though they have not (yet) been dissociated from their original port.

### 5.3. Signaling Protocols

There are multiple protocols that can be used to signal the above messages. One could invent a new protocol for this purpose, or reuse existing protocols, among them LLDP, XMPP, HTTP REST, and VDP [VDP], a new protocol standardized for the purposes of signaling a VM's network parameters from server to l-NVE. Multiple factors influence the choice of protocol(s); this draft's focus is on what needs to be signaled, leaving choices of how the information is signaled, and specific encodings for other drafts to consider.

## 6. Interfacing with DCVPN Control Planes

The control plane for a DCVPN manages the creation/deletion, membership and span of the DCVPN ([I-D.ietf-nvo3-overlay-problem-statement],[I-D.kreeger-nvo3-overlay-cp]). Such a control plane needs to work with the server-to-nve signaling in a coordinated manner, to ensure that address changes at a local NVE are reflected appropriately in remote NVEs. The details of such coordination are specified in separate documents.

## 7. Security Considerations



## 8. IANA Considerations

## 9. Acknowledgments

Many thanks to Amit Shukla for his help with the details of EVB and his insight into data center issues. Many thanks to members of the nvo3 WG for their comments, including Yingjie Gu.

## 10. Informative References

## [I-D.ietf-l2vpn-evpn]

Sajassi, A., Aggarwal, R., Henderickx, W., Balus, F., Isaac, A., and J. Uttaro, "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-03 (work in progress), February 2013.

## [I-D.ietf-nvo3-framework]

Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-02 (work in progress), February 2013.

## [I-D.ietf-nvo3-overlay-problem-statement]

Narten, T., Gray, E., Black, D., Dutt, D., Fang, L., Kreeger, L., Napierala, M., and M. Sridharan, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-02 (work in progress), February 2013.

## [I-D.kreeger-nvo3-overlay-cp]

Kreeger, L., Dutt, D., Narten, T., and M. Sridharan, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp-02 (work in progress), October 2012.

## [I-D.mahalingam-dutt-dcops-vxlan]

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-03 (work in progress), February 2013.

## [I-D.sridharan-virtualization-nvgre]

Sridharan, M., Greenberg, A., Venkataramaiah, N., Wang, Y., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-02 (work in progress), February 2013.

- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC4761] Kompella, K. and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.
- [RFC4762] Lasserre, M. and V. Kompella, "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, January 2007.
- [VDP] IEEE, "Edge Virtual Bridging (IEEE Std 802.1Qbg-2012)", July 2012.

## Authors' Addresses

Kireeti Kompella  
Juniper Networks  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: kireeti@juniper.net

Yakov Rekhter  
Juniper Networks  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: yakov@juniper.net

Thomas Morin  
France Telecom - Orange Labs  
2, avenue Pierre Marzin  
Lannion 22307  
France

Email: thomas.morin@orange.com

David L. Black  
EMC Corporation  
176 South St.  
Hopkinton, MA 01748

Email: david.black@emc.com

Internet Engineering Task Force  
Internet Draft  
Intended status: Informational  
Expires: January 2013

Marc Lasserre  
Florin Balus  
Alcatel-Lucent

Thomas Morin  
France Telecom Orange

Nabil Bitar  
Verizon

Yakov Rekhter  
Juniper

July 9, 2012

Framework for DC Network Virtualization  
draft-lasserre-nvo3-framework-03.txt

#### Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2013.

#### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Abstract

Several IETF drafts relate to the use of overlay networks to support large scale virtual data centers. This draft provides a framework for Network Virtualization over L3 (NVO3) and is intended to help plan a set of work items in order to provide a complete solution set. It defines a logical view of the main components with the intention of streamlining the terminology and focusing the solution set.

## Table of Contents

1. Introduction.....	3
1.1. Conventions used in this document.....	4
1.2. General terminology.....	4
1.3. DC network architecture.....	6
1.4. Tenant networking view.....	7
2. Reference Models.....	8
2.1. Generic Reference Model.....	8
2.2. NVE Reference Model.....	10
2.3. NVE Service Types.....	11
2.3.1. L2 NVE providing Ethernet LAN-like service.....	11
2.3.2. L3 NVE providing IP/VRF-like service.....	11
3. Functional components.....	11
3.1. Generic service virtualization components.....	12
3.1.1. Virtual Access Points (VAPs).....	12
3.1.2. Virtual Network Instance (VNI).....	12
3.1.3. Overlay Modules and VN Context.....	13
3.1.4. Tunnel Overlays and Encapsulation options.....	14
3.1.5. Control Plane Components.....	14
3.1.5.1. Auto-provisioning/Service discovery.....	14
3.1.5.2. Address advertisement and tunnel mapping.....	15

3.1.5.3. Tunnel management.....	15
3.2. Service Overlay Topologies.....	16
4. Key aspects of overlay networks.....	16
4.1. Pros & Cons.....	16
4.2. Overlay issues to consider.....	17
4.2.1. Data plane vs Control plane driven.....	17
4.2.2. Coordination between data plane and control plane...	18
4.2.3. Handling Broadcast, Unknown Unicast and Multicast (BUM) traffic.....	18
4.2.4. Path MTU.....	19
4.2.5. NVE location trade-offs.....	19
4.2.6. Interaction between network overlays and underlays..	20
5. Security Considerations.....	21
6. IANA Considerations.....	21
7. References.....	21
7.1. Normative References.....	21
7.2. Informative References.....	21
8. Acknowledgments.....	22

## 1. Introduction

This document provides a framework for Data Center Network Virtualization over L3 tunnels. This framework is intended to aid in standardizing protocols and mechanisms to support large scale network virtualization for data centers.

Several IETF drafts relate to the use of overlay networks for data centers.

[NVOPS] defines the rationale for using overlay networks in order to build large data center networks. The use of virtualization leads to a very large number of communication domains and end systems to cope with.

[OVCPREQ] describes the requirements for a control plane protocol required by overlay border nodes to exchange overlay mappings.

This document provides reference models and functional components of data center overlay networks as well as a discussion of technical issues that have to be addressed in the design of standards and mechanisms for large scale data centers.

### 1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

### 1.2. General terminology

This document uses the following terminology:

NVE: Network Virtualization Edge. It is a network entity that sits on the edge of the NVO3 network. It implements network virtualization functions that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses). An NVE could be implemented as part of a virtual switch within a hypervisor, a physical switch or router, a Network Service Appliance or even be embedded within an End Station.

VN: Virtual Network. This is a virtual L2 or L3 domain that belongs a tenant.

VNI: Virtual Network Instance. This is one instance of a virtual overlay network. Two Virtual Networks are isolated from one another and may use overlapping addresses.

Virtual Network Context or VN Context: Field that is part of the overlay encapsulation header which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE. The egress NVE uses this field to determine the appropriate virtual network context in which to process the packet. This field MAY be an explicit, unique (to the administrative domain) virtual network identifier (VNID) or MAY express the necessary context information in other ways (e.g. a locally significant identifier).

VNID: Virtual Network Identifier. In the case where the VN context has global significance, this is the ID value that is carried in each data packet in the overlay encapsulation that identifies the Virtual Network the packet belongs to.

Underlay or Underlying Network: This is the network that provides the connectivity between NVEs. The Underlying Network can be

completely unaware of the overlay packets. Addresses within the Underlying Network are also referred to as "outer addresses" because they exist in the outer encapsulation. The Underlying Network can use a completely different protocol (and address family) from that of the overlay.

Data Center (DC): A physical complex housing physical servers, network switches and routers, Network Service Appliances and networked storage. The purpose of a Data Center is to provide application and/or compute and/or storage services. One such service is virtualized data center services, also known as Infrastructure as a Service.

Virtual Data Center or Virtual DC: A container for virtualized compute, storage and network services. Managed by a single tenant, a Virtual DC can contain multiple VNs and multiple Tenant End Systems that are connected to one or more of these VNs.

VM: Virtual Machine. Several Virtual Machines can share the resources of a single physical computer server using the services of a Hypervisor (see below definition).

Hypervisor: Server virtualization software running on a physical compute server that hosts Virtual Machines. The hypervisor provides shared compute/memory/storage and network connectivity to the VMs that it hosts. Hypervisors often embed a Virtual Switch (see below).

Virtual Switch: A function within a Hypervisor (typically implemented in software) that provides similar services to a physical Ethernet switch. It switches Ethernet frames between VMs' virtual NICs within the same physical server, or between a VM and a physical NIC card connecting the server to a physical Ethernet switch. It also enforces network isolation between VMs that should not communicate with each other.

Tenant: A customer who consumes virtualized data center services offered by a cloud service provider. A single tenant may consume one or more Virtual Data Centers hosted by the same cloud service provider.

Tenant End System: It defines an end system of a particular tenant, which can be for instance a virtual machine (VM), a non-virtualized server, or a physical appliance.

ELAN: MEF ELAN, multipoint to multipoint Ethernet service



EVPN: Ethernet VPN as defined in [EVPN]

### 1.3. DC network architecture

A generic architecture for Data Centers is depicted in Figure 1:

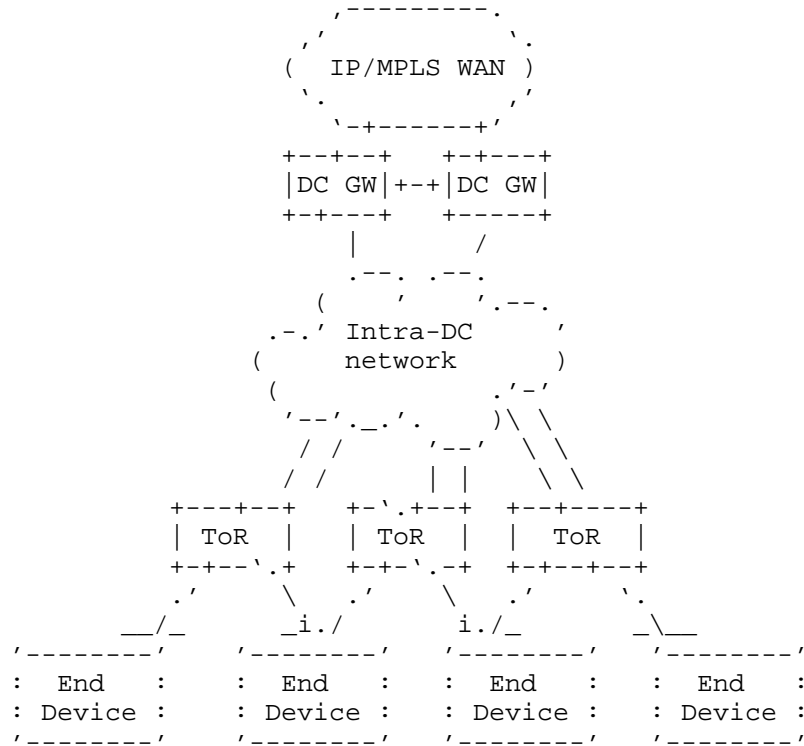


Figure 1 : A Generic Architecture for Data Centers

An example of multi-tier DC network architecture is presented in this figure. It provides a view of physical components inside a DC.

A cloud network is composed of intra-Data Center (DC) networks and network services, and, inter-DC network and network connectivity services. Depending upon the scale, DC distribution, operations model, Capex and Opex aspects, DC networking elements can act as strict L2 switches and/or provide IP routing capabilities, including also service virtualization.

In some DC architectures, it is possible that some tier layers provide L2 and/or L3 services, are collapsed, and that Internet connectivity, inter-DC connectivity and VPN support are handled by a smaller number of nodes. Nevertheless, one can assume that the functional blocks fit with the architecture above.

The following components can be present in a DC:

- o End Device: a DC resource to which the networking service is provided. End Device may be a compute resource (server or server blade), storage component or a network appliance (firewall, load-balancer, IPsec gateway). Alternatively, the End Device may include software based networking functions used to interconnect multiple hosts. An example of soft networking is the virtual switch in the server blades, used to interconnect multiple virtual machines (VMs). End Device may be single or multi-homed to the Top of Rack switches (ToRs).
- o Top of Rack (ToR): Hardware-based Ethernet switch aggregating all Ethernet links from the End Devices in a rack representing the entry point in the physical DC network for the hosts. ToRs may also provide routing functionality, virtual IP network connectivity, or Layer2 tunneling over IP for instance. ToRs are usually multi-homed to switches in the Intra-DC network. Other deployment scenarios may use an intermediate Blade Switch before the ToR or an EoR (End of Row) switch to provide similar function as a ToR.
- o Intra-DC Network: High capacity network composed of core switches aggregating multiple ToRs. Core switches are usually Ethernet switches but can also support routing capabilities.
- o DC GW: Gateway to the outside world providing DC Interconnect and connectivity to Internet and VPN customers. In the current DC network model, this may be simply a Router connected to the Internet and/or an IPVPN/L2VPN PE. Some network implementations may dedicate DC GWs for different connectivity types (e.g., a DC GW for Internet, and another for VPN).

#### 1.4. Tenant networking view

The DC network architecture is used to provide L2 and/or L3 service connectivity to each tenant. An example is depicted in Figure 2:

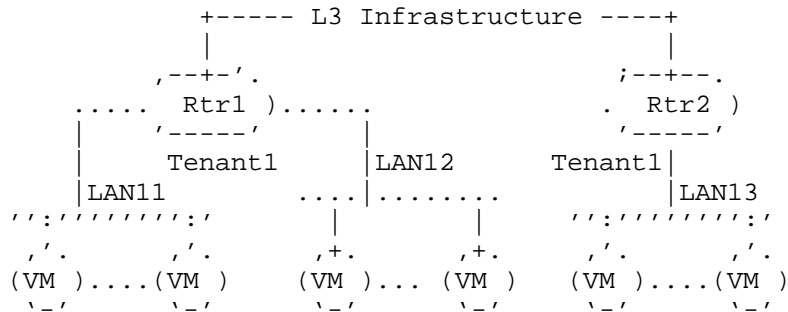


Figure 2 : Logical Service connectivity for a single tenant

In this example one or more L3 contexts and one or more LANs (e.g., one per application type) running on DC switches are assigned for DC tenant 1.

For a multi-tenant DC, a virtualized version of this type of service connectivity needs to be provided for each tenant by the Network Virtualization solution.

## 2. Reference Models

### 2.1. Generic Reference Model

The following diagram shows a DC reference model for network virtualization using Layer3 overlays where edge devices provide a logical interconnect between Tenant End Systems that belong to specific tenant network.

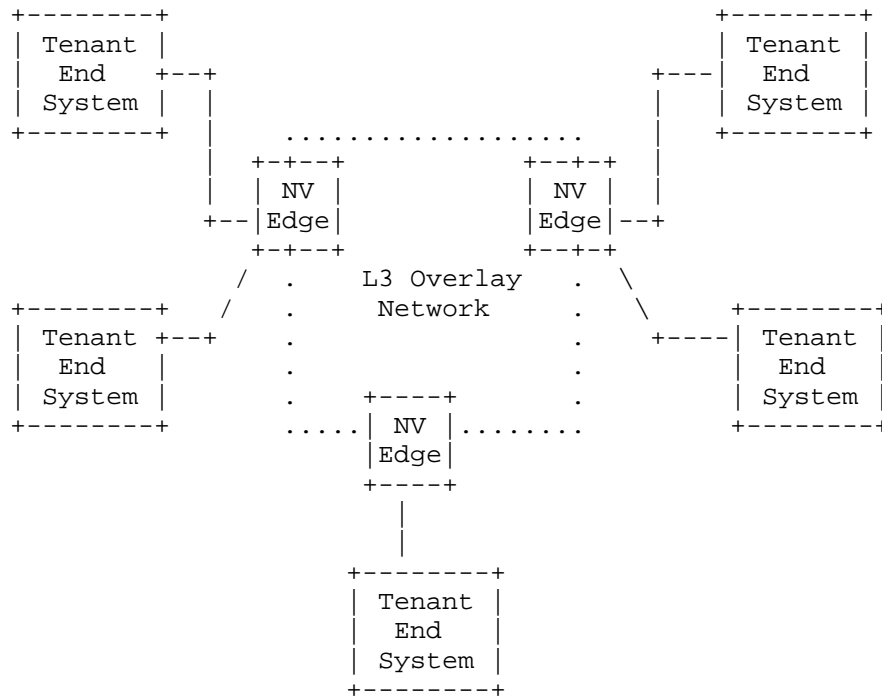


Figure 3 : Generic reference model for DC network virtualization over a Layer3 infrastructure

The functional components in this picture do not necessarily map directly with the physical components described in Figure 1.

For example, an End Device can be a server blade with VMs and virtual switch, i.e. the VM is the Tenant End System and the NVE functions may be performed by the virtual switch and/or the hypervisor.

Another example is the case where an End Device can be a traditional physical server (no VMs, no virtual switch), i.e. the server is the Tenant End System and the NVE functions may be performed by the ToR. Other End Devices in this category are Physical Network Appliances or Storage Systems.

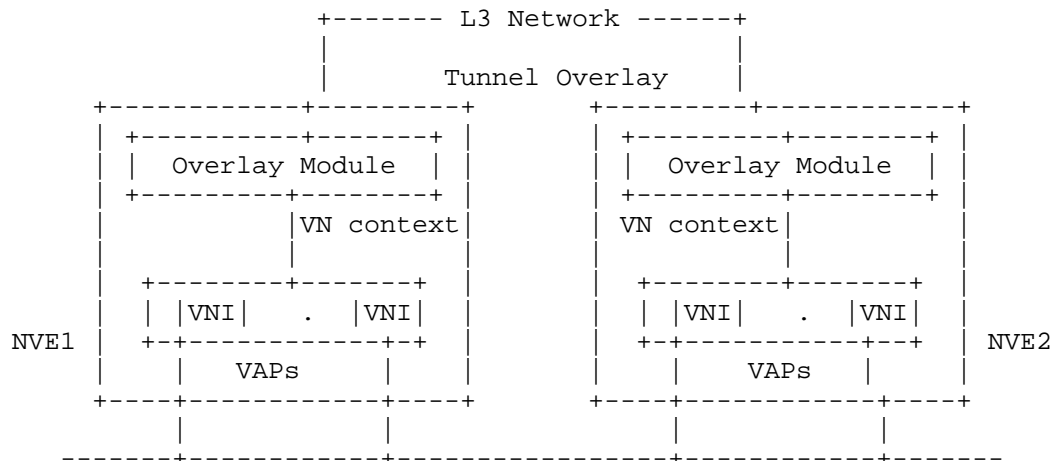
A Tenant End System attaches to a Network Virtualization Edge (NVE) node, either directly or via a switched network (typically Ethernet).

The NVE implements network virtualization functions that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses), tenant-related control plane activity and service contexts from the Routed Backbone nodes.

Core nodes utilize L3 techniques to interconnect NVE nodes in support of the overlay network. These devices perform forwarding based on outer L3 tunnel header, and generally do not maintain per tenant-service state albeit some applications (e.g., multicast) may require control plane or forwarding plane information that pertain to a tenant, group of tenants, tenant service or a set of services that belong to one or more tunnels. When such tenant or tenant-service related information is maintained in the core, overlay virtualization provides knobs to control that information.

## 2.2. NVE Reference Model

The NVE is composed of a tenant service instance that Tenant End Systems interface with and an overlay module that provides tunneling overlay functions (e.g. encapsulation/decapsulation of tenant traffic from/to the tenant forwarding instance, tenant identification and mapping, etc), as described in figure 4:



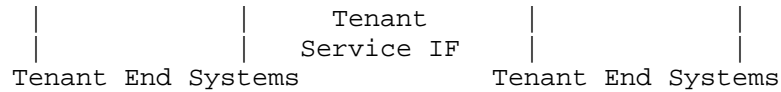


Figure 4 : Generic reference model for NV Edge

Note that some NVE functions (e.g. data plane and control plane functions) may reside in one device or may be implemented separately in different devices.

For example, the NVE functionality could reside solely on the End Devices, on the ToRs or on both the End Devices and the ToRs. In the latter case we say that the the End Device NVE component acts as the NVE Spoke, and ToRs act as NVE hubs. Tenant End Systems will interface with the tenant service instances maintained on the NVE spokes, and tenant service instances maintained on the NVE spokes will interface with the tenant service instances maintained on the NVE hubs.

### 2.3. NVE Service Types

NVE components may be used to provide different types of virtualized service connectivity. This section defines the service types and associated attributes

#### 2.3.1. L2 NVE providing Ethernet LAN-like service

L2 NVE implements Ethernet LAN emulation (ELAN), an Ethernet based multipoint service where the Tenant End Systems appear to be interconnected by a LAN environment over a set of L3 tunnels. It provides per tenant virtual switching instance with MAC addressing isolation and L3 tunnel encapsulation across the core.

#### 2.3.2. L3 NVE providing IP/VRF-like service

Virtualized IP routing and forwarding is similar from a service definition perspective with IETF IP VPN (e.g., BGP/MPLS IPVPN and IPsec VPNs). It provides per tenant routing instance with addressing isolation and L3 tunnel encapsulation across the core.

### 3. Functional components

This section breaks down the Network Virtualization architecture into functional components to make it easier to discuss solution options for different modules.

This version of the document gives an overview of generic functional components that are shared between L2 and L3 service types. Details specific for each service type will be added in future revisions.

### 3.1. Generic service virtualization components

A Network Virtualization solution is built around a number of functional components as depicted in Figure 5:

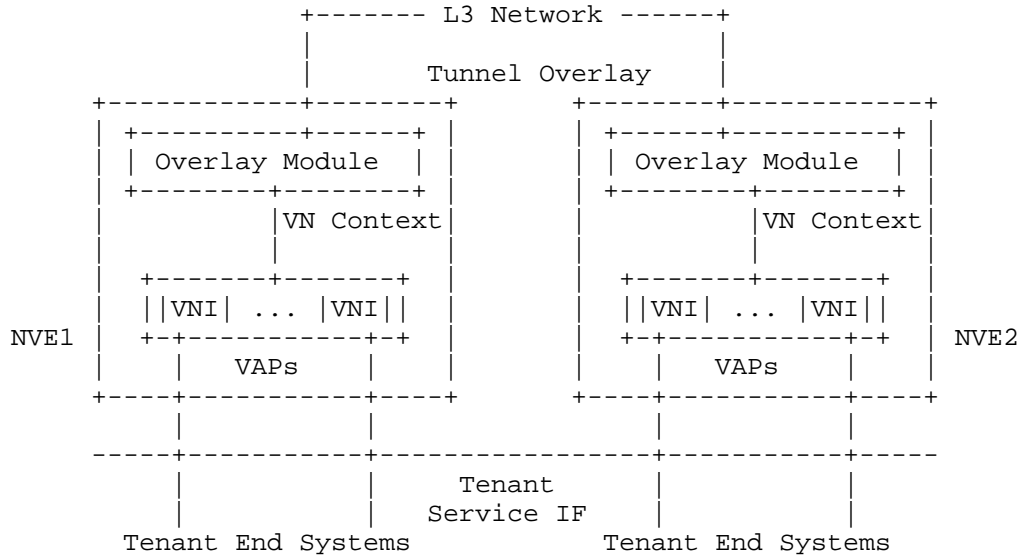


Figure 5 : Generic reference model for NV Edge

#### 3.1.1. Virtual Access Points (VAPs)

Tenant End Systems are connected to the VNI Instance through Virtual Access Points (VAPs). The VAPs can be in reality physical ports on a ToR or virtual ports identified through logical interface identifiers (VLANs, internal VSwitch Interface ID leading to a VM).

#### 3.1.2. Virtual Network Instance (VNI)

The VNI represents a set of configuration attributes defining access and tunnel policies and (L2 and/or L3) forwarding functions.

Per tenant FIB tables and control plane protocol instances are used to maintain separate private contexts between tenants. Hence tenants are free to use their own addressing schemes without concerns about address overlapping with other tenants.

### 3.1.3. Overlay Modules and VN Context

Mechanisms for identifying each tenant service are required to allow the simultaneous overlay of multiple tenant services over the same underlay L3 network topology. In the data plane, each NVE, upon sending a tenant packet, must be able to encode the VN Context for the destination NVE in addition to the L3 tunnel source address identifying the source NVE and the tunnel destination L3 address identifying the destination NVE. This allows the destination NVE to identify the tenant service instance and therefore appropriately process and forward the tenant packet.

The Overlay module provides tunneling overlay functions: tunnel initiation/termination, encapsulation/decapsulation of frames from VAPs/L3 Backbone and may provide for transit forwarding of IP traffic (e.g., transparent tunnel forwarding).

In a multi-tenant context, the tunnel aggregates frames from/to different VNIs. Tenant identification and traffic demultiplexing are based on the VN Context (e.g. VNID).

The following approaches can be considered:

- o One VN Context per Tenant: A globally unique (on a per-DC administrative domain) VNID is used to identify the related Tenant instances. An example of this approach is the use of IEEE VLAN or ISID tags to provide virtual L2 domains.
- o One VN Context per VNI: A per-tenant local value is automatically generated by the egress NVE and usually distributed by a control plane protocol to all the related NVEs. An example of this approach is the use of per VRF MPLS labels in IP VPN [RFC4364].
- o One VN Context per VAP: A per-VAP local value is assigned and usually distributed by a control plane protocol. An example of this approach is the use of per CE-PE MPLS labels in IP VPN [RFC4364].



Note that when using one VN Context per VNI or per VAP, an additional global identifier may be used by the control plane to identify the Tenant context.

#### 3.1.4. Tunnel Overlays and Encapsulation options

Once the VN context is added to the frame, a L3 Tunnel encapsulation is used to transport the frame to the destination NVE. The backbone devices do not usually keep any per service state, simply forwarding the frames based on the outer tunnel header.

Different IP tunneling options (GRE/L2TP/IPSec) and tunneling options (BGP VPN, PW, VPLS) are available for both Ethernet and IP formats.

#### 3.1.5. Control Plane Components

Control plane components may be used to provide the following capabilities:

- . Auto-provisioning/Service discovery
- . Address advertisement and tunnel mapping
- . Tunnel management

A control plane component can be an on-net control protocol or a management control entity.

##### 3.1.5.1. Auto-provisioning/Service discovery

NVEs must be able to select the appropriate VNI for each Tenant End System. This is based on state information that is often provided by external entities. For example, in a VM environment, this information is provided by compute management systems, since these are the only entities that have visibility on which VM belongs to which tenant.

A mechanism for communicating this information between Tenant End Systems and the local NVE is required. As a result the VAPs are created and mapped to the appropriate Tenant Instance.

Depending upon the implementation, this control interface can be implemented using an auto-discovery protocol between Tenant End Systems and their local NVE or through management entities.

When a protocol is used, appropriate security and authentication mechanisms to verify that Tenant End System information is not spoofed or altered are required. This is one critical aspect for providing integrity and tenant isolation in the system.

Another control plane protocol can also be used to advertize NVE tenant service instance (tenant and service type provided to the tenant) to other NVEs. Alternatively, management control entities can also be used to perform these functions.

#### 3.1.5.2. Address advertisement and tunnel mapping

As traffic reaches an ingress NVE, a lookup is performed to determine which tunnel the packet needs to be sent to. It is then encapsulated with a tunnel header containing the destination address of the egress overlay node. Intermediate nodes (between the ingress and egress NVEs) switch or route traffic based upon the outer destination address.

One key step in this process consists of mapping a final destination address to the proper tunnel. NVEs are responsible for maintaining such mappings in their lookup tables. Several ways of populating these lookup tables are possible: control plane driven, management plane driven, or data plane driven.

When a control plane protocol is used to distribute address advertisement and tunneling information, the auto-provisioning/Service discovery could be accomplished by the same protocol. In this scenario, the auto-provisioning/Service discovery could be combined with (be inferred from) the address advertisement and tunnel mapping. Furthermore, a control plane protocol that carries both MAC and IP addresses eliminates the need for ARP, and hence addresses one of the issues with explosive ARP handling.

#### 3.1.5.3. Tunnel management

A control plane protocol may be required to exchange tunnel state information. This may include setting up tunnels and/or providing tunnel state information.

This applies to both unicast and multicast tunnels.

For instance, it may be necessary to provide active/standby status information between NVEs, up/down status information, pruning/grafting information for multicast tunnels, etc.

### 3.2. Service Overlay Topologies

A number of service topologies may be used to optimize the service connectivity and to address NVE performance limitations.

The topology described in Figure 3 suggests the use of a tunnel mesh between the NVEs where each tenant instance is one hop away from a service processing perspective. Partial mesh topologies and an NVE hierarchy may be used where certain NVEs may act as service transit points.

## 4. Key aspects of overlay networks

The intent of this section is to highlight specific issues that proposed overlay solutions need to address.

### 4.1. Pros & Cons

An overlay network is a layer of virtual network topology on top of the physical network.

Overlay networks offer the following key advantages:

- o Unicast tunneling state management is handled at the edge of the network. Intermediate transport nodes are unaware of such state. Note that this is not the case when multicast is enabled in the core network.
- o Tunnels are used to aggregate traffic and hence offer the advantage of minimizing the amount of forwarding state required within the underlay network
- o Decoupling of the overlay addresses (MAC and IP) used by VMs from the underlay network. This offers a clear separation between addresses used within the overlay and the underlay networks and it enables the use of overlapping addresses spaces by Tenant End Systems
- o Support of a large number of virtual network identifiers

Overlay networks also create several challenges:

- o Overlay networks have no controls of underlay networks and lack critical network information

- o Overlays typically probe the network to measure link properties, such as available bandwidth or packet loss rate. It is difficult to accurately evaluate network properties. It might be preferable for the underlay network to expose usage and performance information.
- o Miscommunication between overlay and underlay networks can lead to an inefficient usage of network resources.
- o Fairness of resource sharing and collaboration among end-nodes in overlay networks are two critical issues
- o When multiple overlays co-exist on top of a common underlay network, the lack of coordination between overlays can lead to performance issues.
- o Overlaid traffic may not traverse firewalls and NAT devices.
- o Multicast service scalability. Multicast support may be required in the overlay network to address for each tenant flood containment or efficient multicast handling.
- o Hash-based load balancing may not be optimal as the hash algorithm may not work well due to the limited number of combinations of tunnel source and destination addresses

#### 4.2. Overlay issues to consider

##### 4.2.1. Data plane vs Control plane driven

In the case of an L2NVE, it is possible to dynamically learn MAC addresses against VAPs. It is also possible that such addresses be known and controlled via management or a control protocol for both L2NVEs and L3NVEs.

Dynamic data plane learning implies that flooding of unknown destinations be supported and hence implies that broadcast and/or multicast be supported. Multicasting in the core network for dynamic learning may lead to significant scalability limitations. Specific forwarding rules must be enforced to prevent loops from happening. This can be achieved using a spanning tree, a shortest path tree, or a split-horizon mesh.

It should be noted that the amount of state to be distributed is dependent upon network topology and the number of virtual machines.

Different forms of caching can also be utilized to minimize state distribution between the various elements.

#### 4.2.2. Coordination between data plane and control plane

For an L2 NVE, the NVE needs to be able to determine MAC addresses of the end systems present on a VAP (for instance, dataplane learning may be relied upon for this purpose). For an L3 NVE, the NVE needs to be able to determine IP addresses of the end systems present on a VAP.

In both cases, coordination with the NVE control protocol is needed such that when the NVE determines that the set of addresses behind a VAP has changed, it triggers the local NVE control plane to distribute this information to its peers.

#### 4.2.3. Handling Broadcast, Unknown Unicast and Multicast (BUM) traffic

There are two techniques to support packet replication needed for broadcast, unknown unicast and multicast:

- o Ingress replication
- o Use of core multicast trees

There is a bandwidth vs state trade-off between the two approaches. Depending upon the degree of replication required (i.e. the number of hosts per group) and the amount of multicast state to maintain, trading bandwidth for state is of consideration.

When the number of hosts per group is large, the use of core multicast trees may be more appropriate. When the number of hosts is small (e.g. 2-3), ingress replication may not be an issue.

Depending upon the size of the data center network and hence the number of (S,G) entries, but also the duration of multicast flows, the use of core multicast trees can be a challenge.

When flows are well known, it is possible to pre-provision such multicast trees. However, it is often difficult to predict application flows ahead of time, and hence programming of (S,G) entries for short-lived flows could be impractical.

A possible trade-off is to use in the core shared multicast trees as opposed to dedicated multicast trees.

#### 4.2.4. Path MTU

When using overlay tunneling, an outer header is added to the original frame. This can cause the MTU of the path to the egress tunnel endpoint to be exceeded.

In this section, we will only consider the case of an IP overlay.

It is usually not desirable to rely on IP fragmentation for performance reasons. Ideally, the interface MTU as seen by a Tenant End System is adjusted such that no fragmentation is needed. TCP will adjust its maximum segment size accordingly.

It is possible for the MTU to be configured manually or to be discovered dynamically. Various Path MTU discovery techniques exist in order to determine the proper MTU size to use:

- o Classical ICMP-based MTU Path Discovery [RFC1191] [RFC1981]
  - o Tenant End Systems rely on ICMP messages to discover the MTU of the end-to-end path to its destination. This method is not always possible, such as when traversing middle boxes (e.g. firewalls) which disable ICMP for security reasons
- o Extended MTU Path Discovery techniques such as defined in [RFC4821]

It is also possible to rely on the overlay layer to perform segmentation and reassembly operations without relying on the Tenant End Systems to know about the end-to-end MTU. The assumption is that some hardware assist is available on the NVE node to perform such SAR operations. However, fragmentation by the overlay layer can lead to performance and congestion issues due to TCP dynamics and might require new congestion avoidance mechanisms from the underlay network [FLOYD].

Finally, the underlay network may be designed in such a way that the MTU can accommodate the extra tunnel overhead.

#### 4.2.5. NVE location trade-offs

In the case of DC traffic, traffic originated from a VM is native Ethernet traffic. This traffic can be switched by a local VM switch or ToR switch and then by a DC gateway. The NVE function can be embedded within any of these elements.

There are several criteria to consider when deciding where the NVE processing boundary happens:

- o Processing and memory requirements
  - o Datapath (e.g. lookups, filtering, encapsulation/decapsulation)
  - o Control plane processing (e.g. routing, signaling, OAM)
- o FIB/RIB size
- o Multicast support
  - o Routing protocols
  - o Packet replication capability
- o Fragmentation support
- o QoS transparency
- o Resiliency

#### 4.2.6. Interaction between network overlays and underlays

When multiple overlays co-exist on top of a common underlay network, this can cause some performance issues. These overlays have partially overlapping paths and nodes.

Each overlay is selfish by nature in that it sends traffic so as to optimize its own performance without considering the impact on other overlays, unless the underlay tunnels are traffic engineered on a per overlay basis so as to avoid sharing underlay resources.

Better visibility between overlays and underlays can be achieved by providing mechanisms to exchange information about:

- o Performance metrics (throughput, delay, loss, jitter)
- o Cost metrics

## 5. Security Considerations

The tenant to overlay mapping function can introduce significant security risks if appropriate protocols are not used that can support mutual authentication.

No other new security issues are introduced beyond those described already in the related L2VPN and L3VPN RFCs.

## 6. IANA Considerations

IANA does not need to take any action for this draft.

## 7. References

### 7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

### 7.2. Informative References

[NVOPS] Narten, T. et al, "Problem Statement : Overlays for Network Virtualization", draft-narten-nvo3-overlay-problem-statement (work in progress)

[OVCPREQ] Kreeger, L. et al, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp (work in progress)

[FLOYD] Sally Floyd, Allyn Romanow, "Dynamics of TCP Traffic over ATM Networks", IEEE JSAC, V. 13 N. 4, May 1995

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC1191] Mogul, J. "Path MTU Discovery", RFC1191, November 1990

[RFC1981] McCann, J. et al, "Path MTU Discovery for IPv6", RFC1981, August 1996



[RFC4821] Mathis, M. et al, "Packetization Layer Path MTU  
Discovery", RFC4821, March 2007

## 8. Acknowledgments

In addition to the authors the following people have contributed to this document:

Dimitrios Stiliadis, Rotem Salomonovitch, Alcatel-Lucent

This document was prepared using 2-Word-v2.0.template.dot.

## Authors' Addresses

Marc Lasserre  
Alcatel-Lucent  
Email: marc.lasserre@alcatel-lucent.com

Florin Balus  
Alcatel-Lucent  
777 E. Middlefield Road  
Mountain View, CA, USA 94043  
Email: florin.balus@alcatel-lucent.com

Thomas Morin  
France Telecom Orange  
Email: thomas.morin@orange.com

Nabil Bitar  
Verizon  
40 Sylvan Road  
Waltham, MA 02145  
Email: nabil.bitar@verizon.com

Yakov Rekhter  
Juniper  
Email: yakov@juniper.net

Network working group  
Internet Draft  
Category: Informational

L. Yong  
Huawei  
M. Toy  
Comcast  
A. Isaac  
Bloomberg  
V. Manral  
Hewlett-Packard  
L. Dunbar  
Huawei

Expires: April 2013

October 22, 2012

## Use Cases for DC Network Virtualization Overlays

draft-mity-nvo3-use-case-04

### Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April, 2013.

### Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## Abstract

This draft describes the general NVO3 use cases. The work intention is to help validate the NVO3 framework and requirements as along with the development of the solutions.

## Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

## Table of Contents

1. Introduction.....	3
2. Terminology.....	4
3. Basic Virtual Networks in a Data Center.....	4
4. Interconnecting DC Virtual Network and External Networks.....	6
4.1. DC Virtual Network Access via Internet.....	6
4.2. DC Virtual Network and WAN VPN Interconnection.....	7
5. DC Applications Using NVO3.....	9
5.1. Supporting Multi Technologies in a Data Center.....	9
5.2. Tenant Virtual Network with Bridging/Routing.....	10
5.3. Virtual Data Center (VDC).....	11
5.4. Federating NVO3 Domains.....	13
6. OAM Considerations.....	13
7. Summary.....	13
8. Security Considerations.....	14
9. IANA Considerations.....	14
10. Acknowledgements.....	14
11. References.....	15
11.1. Normative References.....	15
11.2. Informative References.....	15
Authors' Addresses.....	16

## 1. Introduction

Compute Virtualization has dramatically and quickly changed IT industry in terms of efficiency, cost, and the speed in providing a new applications and/or services. However the problems in today's data center hinder the support of an elastic cloud service and dynamic virtual tenant networks [NVO3PRBM]. The goal of DC Network Virtualization Overlays, i.e. NVO3, is to decouple a communication among tenant end systems (VMs) from DC physical networks and to allow the network infrastructure to provide: 1) traffic isolation among one virtual network and another; 2) independent address space in each virtual network and address isolation from the infrastructure's; 3) Flexible VM placement and move from one server to another without any physical network limitation. These characteristics will help address the issues in the data centers.

Although NVO3 may enable a true virtual environment where VMs and net service appliances communicate, the NVO3 solution has to address how to communicate between a virtual network and a physical network. This is because 1) many traditional DCs exist and will not disappear any time soon; 2) a lot of DC applications serve to Internet and/or cooperation users; 3) some applications like Big Data analytics which are CPU bound may not want the virtualization capability.

This document is to describe general NVO3 use cases that apply to various data center networks to ensure nvo3 framework and solutions can meet the demands. Three types of the use cases are:

- o A virtual network connects many tenant end systems within a Data Center and form one L2 or L3 communication domain. A virtual network segregates its traffic from others and allows the VMs in the network moving from one server to another. The case may be used for DC internal applications that constitute the DC East-West traffic.
- o A DC provider offers a secure DC service to an enterprise customer and/or Internet users. In these cases, the enterprise customer may use a traditional VPN provided by a carrier or an IPsec tunnel over Internet connecting to an overlay virtual network offered by a Data Center provider. This is mainly constitutes DC North-South traffic.
- o A DC provider uses NVO3 to design a variety of DC applications that make use of the net service appliance, virtual compute, storage, and networking. In this case, the NVO3 provides the virtual networking functions for the applications.

The document uses the architecture reference model and terminologies defined in [NVO3FRWK] to describe the use cases.

## 2. Terminology

This document uses the terminologies defined in [NVO3FRWK], [RFC4364]. Some additional terms used in the document are listed here.

CUG: Closed User Group

L2 VNI: L2 Virtual Network Instance

L3 VNI: L3 Virtual Network Instance

ARP: Address Resolution Protocol

CPE: Customer Premise Equipment

DNS: Domain Name Service

DMZ: DeMilitarized Zone

NAT: Network Address Translation

VNIF: Internal Virtual Network Interconnection Interface

## 3. Basic Virtual Networks in a Data Center

A virtual network may exist within a DC. The network enables a communication among tenant end systems (TESS) that are in a Closed User Group (CUG). A TES may be a physical server or virtual machine (VM) on a server. A virtual network has a unique virtual network identifier (may be local or global unique) for switches/routers to properly differentiate it from other virtual networks. The CUGs are formed so that proper policies can be applied when the TESS in one CUG communicate with the TESS in other CUGs.

Figure 1 depicts this case by using the framework model. [NVO3FRWK] NVE1 and NVE2 are two network virtual edges and each may exist on a server or ToR. Each NVE may be the member of one or more virtual networks. Each virtual network may be L2 or L3 basis. In this illustration, three virtual networks with VN context Ta, Tn, and Tm are shown. The VN 'Ta' terminates on both NVE1 and NVE2; The VN 'Tn' terminates on NVE1 and the VN 'Tm' at NVE2 only. If an NVE is a member of a VN, one or more virtual network instances (VNI) (i.e. routing and forwarding table) exist on the NVE. Each NVE has one

overlay module to perform frame encapsulation/decapsulation and tunneling initiation/termination. In this scenario, a tunnel between NVE1 and NVE2 is necessary for the virtual network Ta.

A TES attaches to a virtual network (VN) via a virtual access point (VAP) on an NVE. One TES may participate in one or more virtual networks via VAPs; one NVE may be configured with multiple VAPs for a VN. Furthermore if individual virtual networks use different address spaces, the TES participating in all of them will be configured with multiple addresses as well. A TES as a gateway is an example for this. In addition, multiple TESes may use one VAP to attach to a VN. For example, VMS are on a server and NVE is on ToR, some VMS may attach to NVE via one VLAN.

A VNI on an NVE is a routing and forwarding table that caches and/or maintains the mapping of a tenant end system and its attached NVE. The table entry may be updated by the control plane or data plane or management plane. It is possible that an NVE has more than one VNIs associated with a VN.

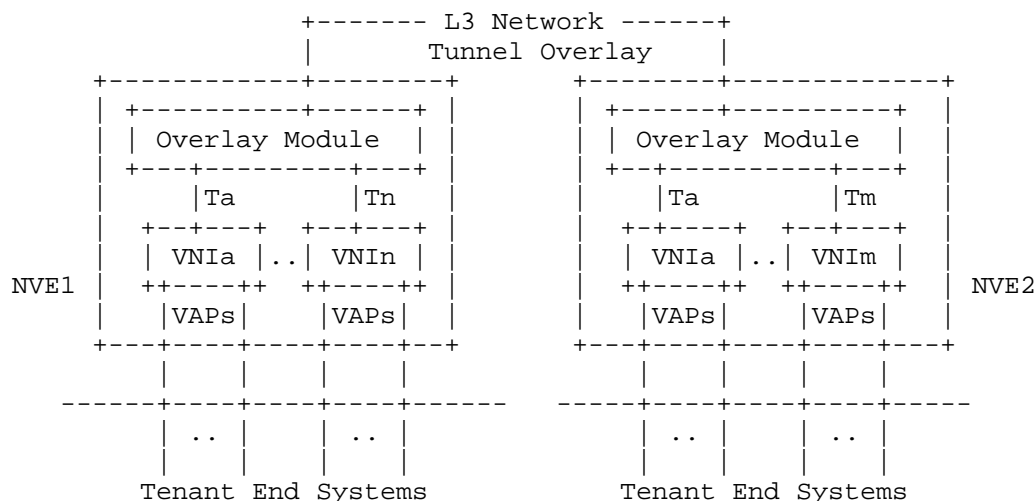


Figure 1 NVo3 for Tenant End-System interconnection

One virtual network may have many NVE members and interconnect several thousands of TESs (as a matter of policy), the capability of supporting a lot of TESs per tenant instance and TES mobility is critical for NVO3 solution no matter where an NVE resides.

It is worth to mention two distinct cases here. The first is when TES and NVE are co-located on a same physical device, which means that the NVE is aware of the TES state at any time via internal API. The second is when TES and NVE are remotely connected, i.e. connected via a switched network or point-to-point link. In this case, a protocol is necessary for NVE to know TES state.

Note that if all NVEs are co-located with TESes in a CUG, the communication in the CUG is in a true virtual environment. If a TES connects to a NVE remotely, the communication from this TES to other TESes in the CUG is not in a true virtual environment. The packets to/from this TES are exposed to a physical network directly, i.e. on a wire.

Individual virtual networks may use its own address space and the space is isolated from DC infrastructure. This eliminates the route changes in the DC underlying network when VMs move. Note that the NVO3 solutions still have to address VM move in the overlay network, i.e. the TES/NVE association change when a VM moves.

If a virtual network spans across multiple DC sites, one design is to allow the corresponding NVO3 instance seamlessly span across those sites without DC gateway routers' termination. In this case, the tunnel between a pair of NVEs may in turn be tunneled over other intermediate tunnels over the Internet or other WANs, or the intra DC and inter DC tunnels are stitched together to form an end-to-end tunnel between two NVEs.

#### 4. Interconnecting DC Virtual Network and External Networks

For customers (an enterprise or individuals) who want to utilize the DC provider's compute and storage resources to run their applications, they need to access those end systems hosted in a DC through Carrier WANs or Internet. A DC provider may want to use an NVO3 virtual network to connect these end systems; then it, in turn, becomes the case of interconnecting DC virtual network and external networks. Two cases are described here.

##### 4.1. DC Virtual Network Access via Internet

A user or an enterprise customer may want to connect to a DC virtual network via Internet but securely. Figure 2 illustrates this case.

An L3 virtual network is configured on NVE1 and NVE2 and two NVEs are connected via an L3 tunnel in the Data Center. A set of tenant end systems attach to NVE1. The NVE2 connects to one (may be more) TES that runs the VN gateway and NAT applications (known as net service appliance). A user or customer can access the VN via Internet by using IPsec tunnel [RFC4301]. The encrypted tunnel is established between the VN GW and the user machine or CPE at enterprise location. The VN GW provides authentication scheme and encryption. Note that VN GW function may be performed by a net service appliance or on a DC GW.

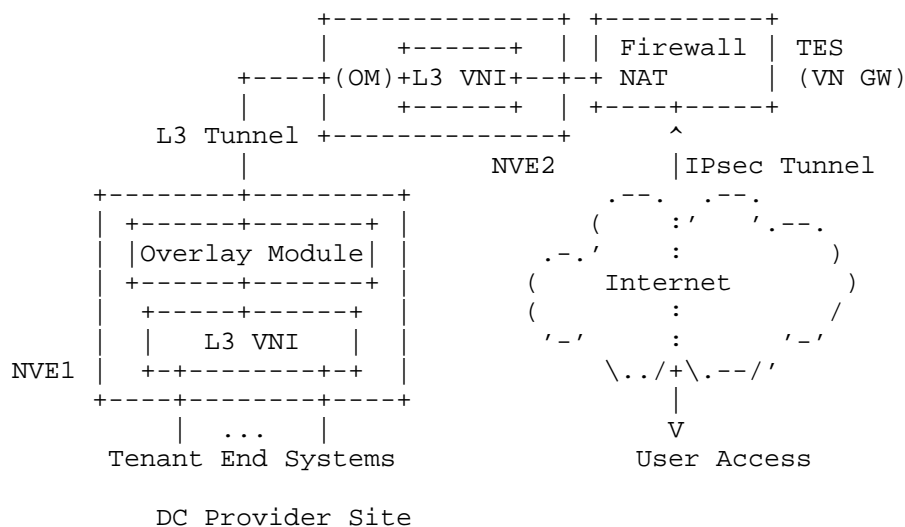


Figure 2 DC Virtual Network Access via Internet

#### 4.2. DC Virtual Network and WAN VPN Interconnection

A DC Provider and Carrier may build a VN and VPN independently and interconnect the two at the DC GW and PE for an enterprise customer. Figure 3 depicts this case in a L3 overlay (L2 overlay is the same). The DC provider constructs an L3 VN between the NVE1 on a server and the NVE2 on the DC GW in the DC site; the carrier constructs an L3VPN between PE1 and PE2 in its IP/MPLS network. An Ethernet Interface physically connects the DC GW and PE2 devices. The local VLAN over the Ethernet interface [VRF-LITE] is configured to connect the L3VNI/NVE2 and VRF, which makes the interconnection between the

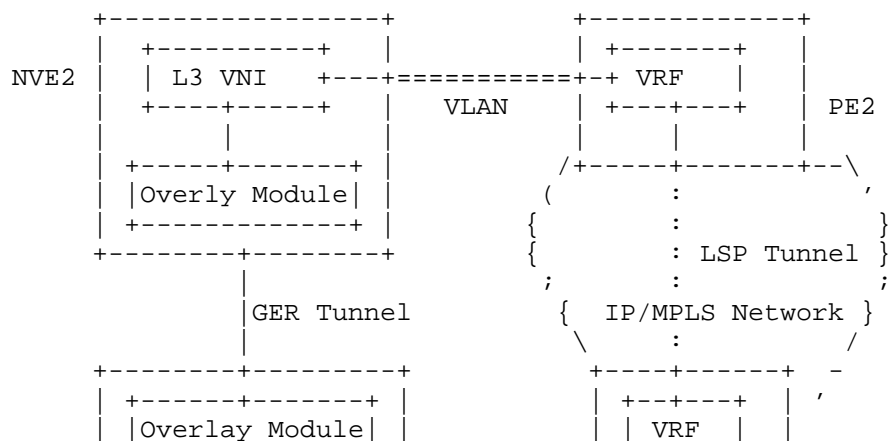


L3 VN in the DC and the L3VPN in IP/MPLS network. An Ethernet Interface may be used between PE1 and CE to connect the L3VPN and enterprise physical networks.

This configuration allows the enterprise networks communicating to the L3 VN as if its own networks but not communicating with DC provider underlying physical networks as well as not other overlay networks in the DC. The enterprise may use its own address space on the L3 VN. The DC provider can manage the VM and storage assignment to the L3 VN for the enterprise customer. The enterprise customer can determine and run their applications on the VMs. From the L3 VN perspective, an end point in the enterprise location appears as the end point associating to the NVE2. The NVE2 on the DC GW has to perform both the GRE tunnel termination [RFC4797] and the local VLAN termination and forward the packets in between. The DC provider and Carrier negotiate the local VLAN ID used on the Ethernet interface.

This configuration makes the L3VPN over the WANs only has the reachability to the TES in the L3 VN. It does not have the reachability of DC physical networks and other VNs in the DC. However, the L3VPN has the reachability of enterprise networks. Note that both the DC provider and enterprise may have multiple network locations connecting to the L3VPN.

The eBGP protocol can be used between DC GW and PE2 for the route population in between. In fact, this is like the Option A in [RFC4364]. This configuration can work with any NVO3 solution. The eBGP, OSPF, or other can be used between PE1 and CE for the route population.



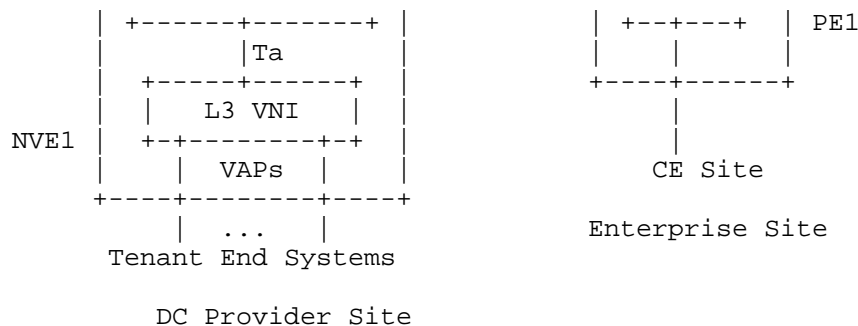


Figure 3 L3 VNI and L3VPN interconnection across multi networks

If an enterprise only has one location, it may use P2P VPWS [RFC4664] or L2TP [RFC5641] to connect one DC provider site. In this case, one edge connects to a physical network and another edge connects to an overlay network.

The interesting feature in this use case is that the L3 VN and compute resource are managed by the DC provider. The DC operator can place them at any location without notifying the enterprise and carrier because the DC physical network is completely isolated from the carrier and enterprise network. Furthermore, the DC operator may move the compute resources assigned to the enterprise from one server to another in the DC without the enterprise customer awareness, i.e. no impact on the enterprise 'live' applications running these resources. Such advanced feature brings some requirements for NVO3 [NVO3PRBM].

## 5. DC Applications Using NVO3

NVO3 brings DC operators the flexibility to design different applications in a true virtual environment without worry about physical network configuration in the Data Center. DC operators may build several virtual networks and interconnect them directly to form a tenant virtual network and implement the communication rules through policy; or may allocate some VMs to run tenant applications and some to run net service applications such as Firewall, DNS for the tenant. Several use cases are given in this section.

### 5.1. Supporting Multi Technologies in a Data Center

Most likely servers deployed in a large data center are rolled in at different times and may have different capacities/features. Some servers may be virtualized, some may not; some may be equipped with

virtual switches, some may not. For the ones equipped with hypervisor based virtual switches, some may support VxLAN [VXLAN] encapsulation, some may support NvGRE encapsulation [NVGRE], and some may not support any types of encapsulation. To construct a tenant virtual network among these servers and the ToRs, it may use two virtual networks and a gateway to allow different implementations working together. For example, one virtual network uses VxLAN encapsulation and another virtual network uses traditional VLAN.

The gateway entity, either on VMs or standalone one, participates in to both virtual networks, and maps the services and identifiers and changes the packet encapsulations.

## 5.2. Tenant Virtual Network with Bridging/Routing

A tenant virtual network may span across multiple Data Centers. DC operator may want to use L2VN within a DC and L3VN outside DCs for a tenant. This is very similar to today's DC physical network configuration. L2 bridging has the simplicity and endpoint awareness while L3 routing has advantages in aggregation and scalability. For this configuration, the virtual gateway function is necessary to interconnect L2VN and L3VN in each DC. Figure 5 illustrates this configuration.

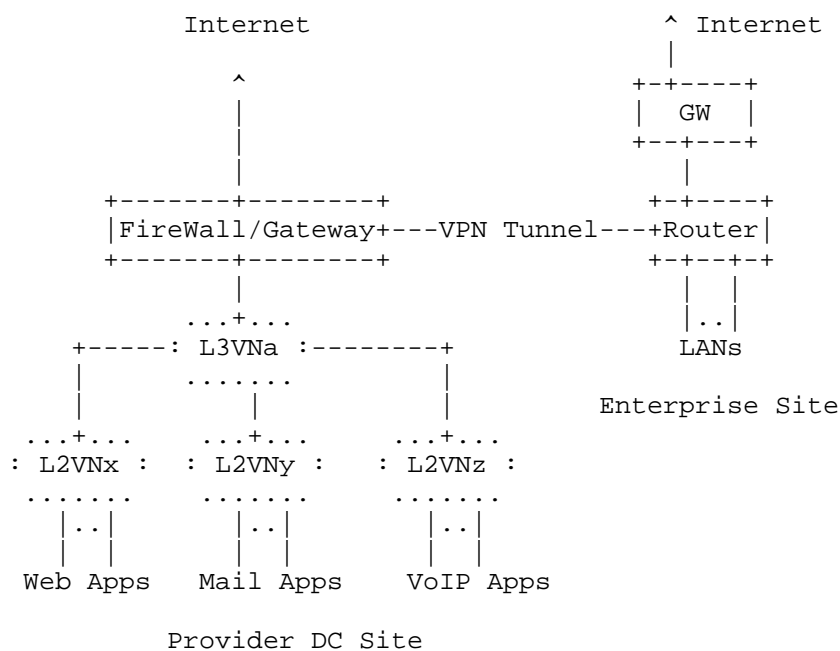
Figure 5 depicts two DC sites. The site A constructs an L2VN that terminates on NVE1, NVE2, and GW1. An L3VN is configured between the GW1 at site A and the GW2 at site Z. An internal Virtual Network Interconnection Interface (VNIF) connects to L2VNI and L3VNI on GW1. Thus the GW1 is the members of the L2VN and L3VN. The L2VNI is the MAC/NVE mapping table and the L3VNI is IP prefix/NVE mapping table. Note that a VNI also has the mapping of TES and VAP at the local NVE. The site Z has the similar configuration. A packet coming to the GW1 from L2VN will be decapsulated and converted into an IP packet and then encapsulated and sent to the site Z. The Gateway uses ARP protocol to obtain MAC/IP mapping. Note that both the L2VN and L3VN in the figure are carried by the tunnels supported by the underlying networks which are not shown in the figure.

Figure 4 Tenant Virtual Network with Bridging/Routing

Enterprise Web/Mail/VoIP applications at the provider DC site; lets the users at Enterprise site to access the applications via the VPN tunnel and Internet via a gateway at the Enterprise site; let Internet users access the applications via the gateway in the provider DC. The enterprise operators can also use the VPN tunnel or IPsec over Internet to access the vDC for the management purpose. The firewall/gateway provides application-level and packet-level gateway function and/or NAT function.

The Enterprise customer decides which applications are accessed by intranet only and which by both intranet and extranet; DC operators then design and configure the proper security policy and gateway function. DC operators may further set different QoS levels for the different applications for a customer.

This application requires the NV03 solution to provide the DC operator an easy way to create NVEs and VNIs for any design and to quickly assign TESSs to a VNI, and easily configure policies on an NVE.



\* firewall/gateway may run on a server or VMs

Figure 5 Virtual Data Center by Using NVO3

#### 5.4. Federating NVO3 Domains

Two general cases are 1) Federating AS managed by a single operator; 2) Federating AS managed by different Operators. The detail will be described in next version.

#### 6. OAM Considerations

NVO3 brings the ability for a DC provider to segregate tenant traffic. A DC provider needs to manage and maintain NVO3 instances. Similarly, the tenant needs to be informed about tunnel failures impacting tenant applications.

Various OAM and SOAM tools and procedures are defined in [IEEE 802.1ag, ITU-T Y.1731, RFC4378, RFC5880, ITU-T Y.1564] for L2 and L3 networks, and for user, including continuity check, loopback, link trace, testing, alarms such as AIS/RDI, and on-demand and periodic measurements. These procedures may apply to tenant overlay networks and tenants not only for proactive maintenance, but also to ensure support of Service Level Agreements (SLAs).

As the tunnel traverses different networks, OAM messages need to be translated at the edge of each network to ensure end-to-end OAM.

It is important that failures at lower layers which do not affect NVO3 instance are to be suppressed.

#### 7. Summary

The document describes some basic potential use cases of NVO3. The combination of these cases should give operators flexibility and power to design more sophisticated cases for various purposes.

The main differences between other overlay network technologies and NVO3 is that the client edges of the NVO3 network are individual and virtualized hosts, not network sites or LANs. NVO3 enables these virtual hosts communicating in a true virtual environment without considering physical network configuration.

NVO3 allows individual tenant virtual networks to use their own address space and isolates the space from the network infrastructure. The approach not only segregates the traffic from multi tenants on a common infrastructure but also makes VM placement and move easier.

DC applications are about providing virtual processing/storage, applications, and networking in a secured and virtualized manner, in which the NVO3 is just a portion of an application. NVO3 decouples the applications and DC network infrastructure configuration.

NVO3's underlying network provides the tunneling between NVEs so that two NVEs appear as one hop to each other. Many tunneling technologies can serve this function. The tunneling may in turn be tunneled over other intermediate tunnels over the Internet or other WANs. It is also possible that intra DC and inter DC tunnels are stitched together to form an end-to-end tunnel between two NVEs.

A DC virtual network may be accessed via an external network in a secure way. Many existing technologies can achieve this.

The key requirements for NVO3 are 1) traffic segregation; 2) supporting a large scale number of virtual networks in a common infrastructure; 3) supporting highly distributed virtual network with sparse memberships 3) VM mobility 4) auto or easy to construct a NVE and its associated TES; 5) Security 6) NVO3 Management [NVO3PRBM].

## 8. Security Considerations

Security is a concern. DC operators need to provide a tenant a secured virtual network, which means the tenant traffic isolated from other tenant's and non-tenant VMs not placed into the tenant virtual network; they also need to prevent DC underlying network from any tenant application attacking through the tenant virtual network or one tenant application attacking another tenant application via DC networks. For example, a tenant application attempts to generate a large volume of traffic to overload DC underlying network. The NVO3 solution has to address these issues.

## 9. IANA Considerations

This document does not request any action from IANA.

## 10. Acknowledgements

Authors like to thank Sue Hares, Young Lee, David Black, Pedro Marques, Mike McBride, David McDysan, and Randy Bush for the review, comments, and suggestions.

## 11. References

### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [IEEE 802.1ag] "Virtual Bridged Local Area Networks - Amendment 5: Connectivity Fault Management", December 2007.
- [ITU-T G.8013/Y.1731] OAM Functions and Mechanisms for Ethernet based Networks, 2011.
- [ITU-T Y.1564] "Ethernet service activation test methodology", 2011.
- [RFC4378] Allan, D., Nadeau, T., "A Framework for Multi-Protocol Label Switching (MPLS) Operations and Management (OAM)", RFC4378, February 2006
- [RFC4301] Kent, S., "Security Architecture for the Internet Protocol", rfc4301, December 2005
- [RFC4664] Andersson, L., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", rfc4664, September 2006
- [RFC4797] Rekhter, Y., etc, "Use of Provider Edge to Provider Edge (PE-PE) Generic Routing Encapsulation (GRE) or IP in BGP/MPLS IP Virtual Private Networks", RFC4797, January 2007
- [RFC5641] McGill, N., "Layer 2 Tunneling Protocol Version 3 (L2TPv3) Extended Circuit Status Values", rfc5641, April 2009.
- [RFC5880] Katz, D. and Ward, D., "Bidirectional Forwarding Detection (BFD)", rfc5880, June 2010.

### 11.2. Informative References

- [NVGRE] Sridharan, M., "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-01, July 2012



[NVO3PRBM] Narten, T., etc "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-00, September 2012

[NVO3FRWK] Lasserre, M., Motin, T., and etc, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-01, October 2012

[VRF-LITE] Cisco, "Configuring VRF-lite", <http://www.cisco.com>

[VXLAN] Mahalingam, M., Dutt, D., etc "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-02.txt, August 2012

#### Authors' Addresses

Lucy Yong  
Huawei Technologies,  
4320 Legacy Dr.  
Plano, Tx75025 US

Phone: +1-469-277-5837  
Email: [lucy.yong@huawei.com](mailto:lucy.yong@huawei.com)

Mehmet Toy  
Comcast  
1800 Bishops Gate Blvd.,  
Mount Laurel, NJ 08054

Phone : +1-856-792-2801  
E-mail : [mehmet\\_toy@cable.comcast.com](mailto:mehmet_toy@cable.comcast.com)

Aldrin Isaac  
Bloomberg  
E-mail: [aldrin.isaac@gmail.com](mailto:aldrin.isaac@gmail.com)

Vishwas Manral  
Hewlett-Packard Corp.  
191111 Pruneridge Ave.  
Cupertino, CA 95014

Phone: 408-447-1497  
Email: [vishwas.manral@hp.com](mailto:vishwas.manral@hp.com)

Linda Dunbar  
Huawei Technologies,  
4320 Legacy Dr.  
Plano, Tx75025 US

Phone: +1-469-277-5840  
Email: linda.dunbar@huawei.com



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: February 11, 2013

T. Narten, Ed.  
IBM  
D. Black  
EMC  
D. Dutt  
  
L. Fang  
Cisco Systems  
E. Gray  
Ericsson  
L. Kreeger  
Cisco  
M. Napierala  
AT&T  
M. Sridharan  
Microsoft  
August 10, 2012

Problem Statement: Overlays for Network Virtualization  
draft-narten-nvo3-overlay-problem-statement-04

## Abstract

This document describes issues associated with providing multi-tenancy in large data center networks that require an overlay-based network virtualization approach to addressing them. A key multi-tenancy requirement is traffic isolation, so that a tenant's traffic is not visible to any other tenant. This isolation can be achieved by assigning one or more virtual networks to each tenant such that traffic within a virtual network is isolated from traffic in other virtual networks. The primary functionality required is provisioning virtual networks, associating a virtual machine's virtual network interface(s) with the appropriate virtual network, and maintaining that association as the virtual machine is activated, migrated and/or deactivated. Use of an overlay-based approach enables scalable deployment on large network infrastructures.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 11, 2013.

#### Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
2. Problem Areas . . . . .	5
2.1. Need For Dynamic Provisioning . . . . .	5
2.2. Virtual Machine Mobility Limitations . . . . .	6
2.3. Inadequate Forwarding Table Sizes in Switches . . . . .	6
2.4. Need to Decouple Logical and Physical Configuration . . . . .	7
2.5. Need For Address Separation Between Tenants . . . . .	7
2.6. Need For Address Separation Between Tenant and Infrastructure . . . . .	7
2.7. IEEE 802.1 VLAN Limitations . . . . .	8
3. Network Overlays . . . . .	8
3.1. Benefits of Network Overlays . . . . .	9
3.2. Communication Between Virtual and Traditional Networks . . . . .	10
3.3. Communication Between Virtual Networks . . . . .	11
3.4. Overlay Design Characteristics . . . . .	11
3.5. Overlay Networking Work Areas . . . . .	12
4. Related IETF and IEEE Work . . . . .	14
4.1. L3 BGP/MPLS IP VPNs . . . . .	14
4.2. L2 BGP/MPLS IP VPNs . . . . .	15
4.3. IEEE 802.1aq - Shortest Path Bridging . . . . .	15
4.4. ARMD . . . . .	15
4.5. TRILL . . . . .	15
4.6. L2VPNs . . . . .	16
4.7. Proxy Mobile IP . . . . .	16
4.8. LISP . . . . .	16
5. Further Work . . . . .	16
6. Summary . . . . .	17
7. Acknowledgments . . . . .	17
8. IANA Considerations . . . . .	17
9. Security Considerations . . . . .	17
10. Informative References . . . . .	17
Appendix A. Change Log . . . . .	19
A.1. Changes from -01 . . . . .	19
A.2. Changes from -02 . . . . .	19
A.3. Changes from -03 . . . . .	20
Authors' Addresses . . . . .	20

## 1. Introduction

Data Centers are increasingly being consolidated and outsourced in an effort, both to improve the deployment time of applications as well as reduce operational costs. This coincides with an increasing demand for compute, storage, and network resources from applications. In order to scale compute, storage, and network resources, physical resources are being abstracted from their logical representation, in what is referred to as server, storage, and network virtualization. Virtualization can be implemented in various layers of computer systems or networks

The demand for server virtualization is increasing in data centers. With server virtualization, each physical server supports multiple virtual machines (VMs), each running its own operating system, middleware and applications. Virtualization is a key enabler of workload agility, i.e., allowing any server to host any application and providing the flexibility of adding, shrinking, or moving services within the physical infrastructure. Server virtualization provides numerous benefits, including higher utilization, increased security, reduced user downtime, reduced power usage, etc.

Multi-tenant data centers are taking advantage of the benefits of server virtualization to provide a new kind of hosting, a virtual hosted data center. Multi-tenant data centers are ones where individual tenants could belong to a different company (in the case of a public provider) or a different department (in the case of an internal company data center). Each tenant has the expectation of a level of security and privacy separating their resources from those of other tenants. For example, one tenant's traffic must never be exposed to another tenant, except through carefully controlled interfaces, such as a security gateway.

To a tenant, virtual data centers are similar to their physical counterparts, consisting of end stations attached to a network, complete with services such as load balancers and firewalls. But unlike a physical data center, end stations connect to a virtual network. To end stations, a virtual network looks like a normal network (e.g., providing an ethernet or L3 service), except that the only end stations connected to the virtual network are those belonging to a tenant's specific virtual network.

A tenant is the administrative entity that is responsible for and manages a specific virtual network instance and its associated services (whether virtual or physical). In a cloud environment, a tenant would correspond to the customer that has defined and is using a particular virtual network. However, a tenant may also find it useful to create multiple different virtual network instances.

Hence, there is a one-to-many mapping between tenants and virtual network instances. A single tenant may operate multiple individual virtual network instances, each associated with a different service.

How a virtual network is implemented does not generally matter to the tenant; what matters is that the service provided (L2 or L3) has the right semantics, performance, etc. It could be implemented via a pure routed network, a pure bridged network or a combination of bridged and routed networks. A key requirement is that each individual virtual network instance be isolated from other virtual network instances.

For data center virtualization, two key issues must be addressed. First, address space separation between tenants must be supported. Second, it must be possible to place (and migrate) VMs anywhere in the data center, without restricting VM addressing to match the subnet boundaries of the underlying data center network.

This document outlines the problems encountered in scaling the number of isolated networks in a data center, as well as the problems of managing the creation/deletion, membership and span of these networks and makes the case that an overlay based approach, where individual networks are implemented within individual virtual networks that are dynamically controlled by a standardized control plane provides a number of advantages over current approaches. The purpose of this document is to identify the set of problems that any solution has to address in building multi-tenant data centers. With this approach, the goal is to allow the construction of standardized, interoperable implementations to allow the construction of multi-tenant data centers.

Section 2 describes the problem space details. Section 3 describes overlay networks in more detail. Sections 4 and 5 review related and further work, while Section 6 closes with a summary.

## 2. Problem Areas

The following subsections describe aspects of multi-tenant data center networking that pose problems for network infrastructure. Different problem aspects may arise based on the network architecture and scale.

### 2.1. Need For Dynamic Provisioning

Cloud computing involves on-demand provisioning of resources for multi-tenant environments. A common example of cloud computing is the public cloud, where a cloud service provider offers elastic



services to multiple customers over the same infrastructure. In current systems, it can be difficult to provision resources for individual tenants in such a way that provisioned properties migrate automatically when services are dynamically moved around within the data center to optimize workloads.

## 2.2. Virtual Machine Mobility Limitations

A key benefit of server virtualization is virtual machine (VM) mobility. A VM can be migrated from one server to another, live, i.e., while continuing to run and without needing to shut it down and restart it at the new location. A key requirement for live migration is that a VM retain critical network state at its new location, including its IP and MAC address(es). Preservation of MAC addresses may be necessary, for example, when software licenses are bound to MAC addresses. More generally, any change in the VM's MAC addresses resulting from a move would be visible to the VM and thus potentially result in unexpected disruptions. Retaining IP addresses after a move is necessary to prevent existing transport connections (e.g., TCP) from breaking and needing to be restarted.

In traditional data centers, servers are assigned IP addresses based on their physical location, for example based on the Top of Rack (ToR) switch for the server rack or the VLAN configured to the server. Servers can only move to other locations within the same IP subnet. This constraint is not problematic for physical servers, which move infrequently, but it restricts the placement and movement of VMs within the data center. Any solution for a scalable multi-tenant data center must allow a VM to be placed (or moved) anywhere within the data center, without being constrained by the subnet boundary concerns of the host servers.

## 2.3. Inadequate Forwarding Table Sizes in Switches

Today's virtualized environments place additional demands on the forwarding tables of switches in the physical infrastructure. Instead of just one link-layer address per server, the switching infrastructure has to learn addresses of the individual VMs (which could range in the 100s per server). This is a requirement since traffic from/to the VMs to the rest of the physical network will traverse the physical network infrastructure. This places a much larger demand on the switches' forwarding table capacity compared to non-virtualized environments, causing more traffic to be flooded or dropped when the number of addresses in use exceeds a switch's forwarding table capacity.

#### 2.4. Need to Decouple Logical and Physical Configuration

Data center operators must be able to achieve high utilization of server and network capacity. For efficient and flexible allocation, operators should be able to spread a virtual network instance across servers in any rack in the data center. It should also be possible to migrate compute workloads to any server anywhere in the network while retaining the workload's addresses. In networks using VLANs, moving servers elsewhere in the network may require expanding the scope of the VLAN beyond its original boundaries. While this can be done, it requires potentially complex network configuration changes and can conflict with the desire to bound the size of broadcast domains, especially in larger data centers.

However, in order to limit the broadcast domain of each VLAN, multi-destination frames within a VLAN should optimally flow only to those devices that have that VLAN configured. When workloads migrate, the physical network (e.g., access lists) may need to be reconfigured which is typically time consuming and error prone.

An important use case is cross-pod expansion. A pod typically consists of one or more racks of servers with its associated network and storage connectivity. A tenant's virtual network may start off on a pod and, due to expansion, require servers/VMs on other pods, especially the case when other pods are not fully utilizing all their resources. This use case requires that virtual networks span multiple pods in order to provide connectivity to all of its tenant's servers/VMs. Such expansion can be difficult to achieve when tenant addressing is tied to the addressing used by the underlay network or when it requires that the scope of the underlying L2 VLAN expand beyond its original pod boundary.

#### 2.5. Need For Address Separation Between Tenants

Individual tenants need control over the addresses they use within a virtual network. But it can be problematic when different tenants want to use the same addresses, or even if the same tenant wants to reuse the same addresses in different virtual networks. Consequently, virtual networks must allow tenants to use whatever addresses they want without concern for what addresses are being used by other tenants or other virtual networks.

#### 2.6. Need For Address Separation Between Tenant and Infrastructure

As in the previous case, a tenant needs to be able to use whatever addresses it wants in a virtual network independent of what addresses the underlying data center network is using. Tenants (and the underlay infrastructure provider) should be able use whatever

addresses make sense for them, without having to worry about address collisions between addresses used by tenants and those used by the underlay data center network.

## 2.7. IEEE 802.1 VLAN Limitations

VLANs are a well known construct in the networking industry, providing an L2 service via an L2 underlay. A VLAN is an L2 bridging construct that provides some of the semantics of virtual networks mentioned above: a MAC address is unique within a VLAN, but not necessarily across VLANs. Traffic sourced within a VLAN (including broadcast and multicast traffic) remains within the VLAN it originates from. Traffic forwarded from one VLAN to another typically involves router (L3) processing. The forwarding table look up operation is keyed on {VLAN, MAC address} tuples.

But there are problems and limitations with L2 VLANs. VLANs are a pure L2 bridging construct and VLAN identifiers are carried along with data frames to allow each forwarding point to know what VLAN the frame belongs to. A VLAN today is defined as a 12 bit number, limiting the total number of VLANs to 4096 (though typically, this number is 4094 since 0 and 4095 are reserved). Due to the large number of tenants that a cloud provider might service, the 4094 VLAN limit is often inadequate. In addition, there is often a need for multiple VLANs per tenant, which exacerbates the issue. The use of a sufficiently large VNID, present in the overlay control plane and possibly also in the dataplane would eliminate current VLAN size limitations associated with single 12-bit VLAN tags.

## 3. Network Overlays

Virtual Networks are used to isolate a tenant's traffic from that of other tenants (or even traffic within the same tenant that requires isolation). There are two main characteristics of virtual networks:

1. Virtual networks isolate the address space used in one virtual network from the address space used by another virtual network. The same network addresses may be used in different virtual networks at the same time. In addition, the address space used by a virtual network is independent from that used by the underlying physical network.
2. Virtual Networks limit the scope of packets sent on the virtual network. Packets sent by end systems attached to a virtual network are delivered as expected to other end systems on that virtual network and may exit a virtual network only through controlled exit points such as a security gateway. Likewise,

packets sourced from outside of the virtual network may enter the virtual network only through controlled entry points, such as a security gateway.

### 3.1. Benefits of Network Overlays

To address the problems described in Section 2, a network overlay model can be used.

The idea behind an overlay is quite straightforward. Each virtual network instance is implemented as an overlay. The original packet is encapsulated by the first-hop network device. The encapsulation identifies the destination of the device that will perform the decapsulation before delivering the original packet to the endpoint. The rest of the network forwards the packet based on the encapsulation header and can be oblivious to the payload that is carried inside.

Overlays are based on what is commonly known as a "map-and-encap" architecture. There are three distinct and logically separable steps:

1. The first-hop overlay device implements a mapping operation that determines where the encapsulated packet should be sent to reach its intended destination VM. Specifically, the mapping function maps the destination address (either L2 or L3) of a packet received from a VM into the corresponding destination address of the egress device. The destination address will be the underlay address of the device doing the decapsulation and is an IP address.
2. Once the mapping has been determined, the ingress overlay device encapsulates the received packet within an overlay header.
3. The final step is to actually forward the (now encapsulated) packet to its destination. The packet is forwarded by the underlay (i.e., the IP network) based entirely on its outer address. Upon receipt at the destination, the egress overlay device decapsulates the original packet and delivers it to the intended recipient VM.

Each of the above steps is logically distinct, though an implementation might combine them for efficiency or other reasons. It should be noted that in L3 BGP/VPN terminology, the above steps are commonly known as "forwarding" or "virtual forwarding".

The first hop network device can be a traditional switch or router or the virtual switch residing inside a hypervisor. Furthermore, the

endpoint can be a VM or it can be a physical server. Examples of architectures based on network overlays include BGP/MPLS VPNs [RFC4364], TRILL [RFC6325], LISP [I-D.ietf-lisp], and Shortest Path Bridging (SPB-M) [SPBM].

In the data plane, a virtual network identifier (or VNID), or a locally significant identifier, can be carried as part of the overlay header so that every data packet explicitly identifies the specific virtual network the packet belongs to. Since both routed and bridged semantics can be supported by a virtual data center, the original packet carried within the overlay header can be an Ethernet frame complete with MAC addresses or just the IP packet.

The use of a sufficiently large VNID would address current VLAN limitations associated with single 12-bit VLAN tags. This VNID can be carried in the control plane. In the data plane, an overlay header provides a place to carry either the VNID, or an identifier that is locally-significant to the edge device. In both cases, the identifier in the overlay header specifies which virtual network the data packet belongs to.

A key aspect of overlays is the decoupling of the "virtual" MAC and/or IP addresses used by VMs from the physical network infrastructure and the infrastructure IP addresses used by the data center. If a VM changes location, the overlay edge devices simply update their mapping tables to reflect the new location of the VM within the data center's infrastructure space. Because an overlay network is used, a VM can now be located anywhere in the data center that the overlay reaches without regards to traditional constraints implied by L2 properties such as VLAN numbering, or the span of an L2 broadcast domain scoped to a single pod or access switch.

Multi-tenancy is supported by isolating the traffic of one virtual network instance from traffic of another. Traffic from one virtual network instance cannot be delivered to another instance without (conceptually) exiting the instance and entering the other instance via an entity that has connectivity to both virtual network instances. Without the existence of this entity, tenant traffic remains isolated within each individual virtual network instance.

Overlays are designed to allow a set of VMs to be placed within a single virtual network instance, whether that virtual network provides a bridged network or a routed network.

### 3.2. Communication Between Virtual and Traditional Networks

Not all communication will be between devices connected to virtualized networks. Devices using overlays will continue to access

devices and make use of services on traditional, non-virtualized networks, whether in the data center, the public Internet, or at remote/branch campuses. Any virtual network solution must be capable of interoperating with existing routers, VPN services, load balancers, intrusion detection services, firewalls, etc. on external networks.

Communication between devices attached to a virtual network and devices connected to non-virtualized networks is handled architecturally by having specialized gateway devices that receive packets from a virtualized network, decapsulate them, process them as regular (i.e., non-virtualized) traffic, and finally forward them on to their appropriate destination (and vice versa). Additional identification, such as VLAN tags, could be used on the non-virtualized side of such a gateway to enable forwarding of traffic for multiple virtual networks over a common non-virtualized link.

A wide range of implementation approaches are possible. Overlay gateway functionality could be combined with other network functionality into a network device that implements the overlay functionality, and then forwards traffic between other internal components that implement functionality such as full router service, load balancing, firewall support, VPN gateway, etc.

### 3.3. Communication Between Virtual Networks

Communication between devices on different virtual networks is handled architecturally by adding specialized interconnect functionality among the otherwise isolated virtual networks. For a virtual network providing an L2 service, such interconnect functionality could be IP forwarding configured as part of the "default gateway" for each virtual network. For a virtual network providing L3 service, the interconnect functionality could be IP forwarding configured as part of routing between IP subnets or it can be based on configured inter-virtual network traffic policies. In both cases, the implementation of the interconnect functionality could be distributed across the NVEs, and could be combined with other network functionality (e.g., load balancing, firewall support) that is applied to traffic that is forwarded between virtual networks.

### 3.4. Overlay Design Characteristics

There are existing layer 2 and layer 3 overlay protocols in existence, but they do not necessarily solve all of today's problem in the environment of a highly virtualized data center. Below are some of the characteristics of environments that must be taken into account by the overlay technology:

1. Highly distributed systems. The overlay should work in an environment where there could be many thousands of access devices (e.g. residing within the hypervisors) and many more end systems (e.g. VMs) connected to them. This leads to a distributed mapping system that puts a low overhead on the overlay tunnel endpoints.
2. Many highly distributed virtual networks with sparse membership. Each virtual network could be highly dispersed inside the data center. Also, along with expectation of many virtual networks, the number of end systems connected to any one virtual network is expected to be relatively low; Therefore, the percentage of access devices participating in any given virtual network would also be expected to be low. For this reason, efficient delivery of multi-destination traffic within a virtual network instance should be taken into consideration.
3. Highly dynamic end systems. End systems connected to virtual networks can be very dynamic, both in terms of creation/deletion/power-on/off and in terms of mobility across the access devices.
4. Work with existing, widely deployed network Ethernet switches and IP routers without requiring wholesale replacement. The first hop device (or end system) that adds and removes the overlay header will require new equipment and/or new software.
5. Work with existing data center network deployments without requiring major changes in operational or other practices. For example, some data centers have not enabled multicast beyond link-local scope. Overlays should be capable of leveraging underlay multicast support where appropriate, but not require its enablement in order to use an overlay solution.
6. Network infrastructure administered by a single administrative domain. This is consistent with operation within a data center, and not across the Internet.

### 3.5. Overlay Networking Work Areas

There are three specific and separate potential work areas needed to realize an overlay solution. The areas correspond to different possible "on-the-wire" protocols, where distinct entities interact with each other.

One area of work concerns the address dissemination protocol an NVE uses to build and maintain the mapping tables it uses to deliver encapsulated packets to their proper destination. One approach is to build mapping tables entirely via learning (as is done in 802.1

networks). But to provide better scaling properties, a more sophisticated approach is needed, i.e., the use of a specialized control plane protocol. While there are some advantages to using or leveraging an existing protocol for maintaining mapping tables, the fact that large numbers of NVE's will likely reside in hypervisors places constraints on the resources (cpu and memory) that can be dedicated to such functions. For example, routing protocols (e.g., IS-IS, BGP) may have scaling difficulties if implemented directly in all NVEs, based on both flooding and convergence time concerns. An alternative approach would be to use a standard query protocol between NVEs and the set of network nodes that maintain address mappings used across the data center for the entire overlay system.

From an architectural perspective, one can view the address mapping dissemination problem as having two distinct and separable components. The first component consists of a back-end "oracle" that is responsible for distributing and maintaining the mapping information for the entire overlay system. The second component consists of the on-the-wire protocols an NVE uses when interacting with the oracle.

The back-end oracle could provide high performance, high resiliency, failover, etc. and could be implemented in significantly different ways. For example, one model uses a traditional, centralized "directory-based" database, using replicated instances for reliability and failover. A second model involves using and possibly extending an existing routing protocol (e.g., BGP, IS-IS, etc.). To support different architectural models, it is useful to have one standard protocol for the NVE-oracle interaction while allowing different protocols and architectural approaches for the oracle itself. Separating the two allows NVEs to transparently interact with different types of oracles, i.e., either of the two architectural models described above. Having separate protocols could also allow for a simplified NVE that only interacts with the oracle for the mapping table entries it needs and allows the oracle (and its associated protocols) to evolve independently over time with minimal impact to the NVEs.

A third work area considers the attachment and detachment of VMs (or Tenant End Systems [I-D.lasserre-nvo3-framework] more generally) from a specific virtual network instance. When a VM attaches, the Network Virtualization Edge (NVE) [I-D.lasserre-nvo3-framework] associates the VM with a specific overlay for the purposes of tunneling traffic sourced from or destined to the VM. When a VM disconnects, it is removed from the overlay and the NVE effectively terminates any tunnels associated with the VM. To achieve this functionality, a standardized interaction between the NVE and hypervisor may be needed, for example in the case where the NVE resides on a separate



device from the VM.

In summary, there are three areas of potential work. The first area concerns the oracle itself and any on-the-wire protocols it needs. A second area concerns the interaction between the oracle and NVEs. The third work area concerns protocols associated with attaching and detaching a VM from a particular virtual network instance. All three work areas are important to the development of scalable, interoperable solutions.

#### 4. Related IETF and IEEE Work

The following subsections discuss related IETF and IEEE work in progress, the items are not meant to be complete coverage of all IETF and IEEE data center related work, nor are the descriptions comprehensive. Each area is currently trying to address certain limitations of today's data center networks, e.g., scaling is a common issue for every area listed and multi-tenancy and VM mobility are important focus areas as well. Comparing and evaluating the work result and progress of each work area listed is out of scope of this document. The intent of this section is to provide a reference to the interested readers.

##### 4.1. L3 BGP/MPLS IP VPNs

BGP/MPLS IP VPNs [RFC4364] support multi-tenancy address overlapping, VPN traffic isolation, and address separation between tenants and network infrastructure. The BGP/MPLS control plane is used to distribute the VPN labels and the tenant IP addresses which identify the tenants (or to be more specific, the particular VPN/VN) and tenant IP addresses. Deployment of enterprise L3 VPNs has been shown to scale to thousands of VPNs and millions of VPN prefixes. BGP/MPLS IP VPNs are currently deployed in some large enterprise data centers. The potential limitation for deploying BGP/MPLS IP VPNs in data center environments is the practicality of using BGP in the data center, especially reaching into the servers or hypervisors. There may be computing work force skill set issues, equipment support issues, and potential new scaling challenges. A combination of BGP and lighter weight IP signaling protocols, e.g., XMPP, have been proposed to extend the solutions into DC environment [I-D.margues-end-system], while taking advantage of building in VPN features with its rich policy support; it is especially useful for inter-tenant connectivity.

#### 4.2. L2 BGP/MPLS IP VPNs

Ethernet Virtual Private Networks (E-VPNs) [I-D.ietf-l2vpn-evpn] provide an emulated L2 service in which each tenant has its own Ethernet network over a common IP or MPLS infrastructure and a BGP/MPLS control plane is used to distribute the tenant MAC addresses and the MPLS labels that identify the tenants and tenant MAC addresses. Within the BGP/MPLS control plane a thirty two bit Ethernet Tag is used to identify the broadcast domains (VLANs) associated with a given L2 VLAN service instance and these Ethernet tags are mapped to VLAN IDs understood by the tenant at the service edges. This means that the limit of 4096 VLANs is associated with an individual tenant service edge, enabling a much higher level of scalability. Interconnection between tenants is also allowed in a controlled fashion.

VM Mobility [I-D.raggarwa-data-center-mobility] introduces the concept of a combined L2/L3 VPN service in order to support the mobility of individual Virtual Machines (VMs) between Data Centers connected over a common IP or MPLS infrastructure.

#### 4.3. IEEE 802.1aq - Shortest Path Bridging

Shortest Path Bridging (SPB-M) is an IS-IS based overlay for L2 Ethernet. SPB-M supports multi-pathing and addresses a number of shortcoming in the original Ethernet Spanning Tree Protocol. SPB-M uses IEEE 802.1ah MAC-in-MAC encapsulation and supports a 24-bit I-SID, which can be used to identify virtual network instances. SPB-M is entirely L2 based, extending the L2 Ethernet bridging model.

#### 4.4. ARMD

ARMD is chartered to look at data center scaling issues with a focus on address resolution. ARMD is currently chartered to develop a problem statement and is not currently developing solutions. While an overlay-based approach may address some of the "pain points" that have been raised in ARMD (e.g., better support for multi-tenancy), an overlay approach may also push some of the L2 scaling concerns (e.g., excessive flooding) to the IP level (flooding via IP multicast). Analysis will be needed to understand the scaling tradeoffs of an overlay based approach compared with existing approaches. On the other hand, existing IP-based approaches such as proxy ARP may help mitigate some concerns.

#### 4.5. TRILL

TRILL is an L2-based approach aimed at improving deficiencies and limitations with current Ethernet networks and STP in particular.

Although it differs from Shortest Path Bridging in many architectural and implementation details, it is similar in that it provides an L2-based service to end systems. TRILL as defined today, supports only the standard (and limited) 12-bit VLAN model. Approaches to extend TRILL to support more than 4094 VLANs are currently under investigation [I-D.ietf-trill-fine-labeling]

#### 4.6. L2VPNs

The IETF has specified a number of approaches for connecting L2 domains together as part of the L2VPN Working Group. That group, however has historically been focused on Provider-provisioned L2 VPNs, where the service provider participates in management and provisioning of the VPN. In addition, much of the target environment for such deployments involves carrying L2 traffic over WANs. Overlay approaches are intended to be used within data centers where the overlay network is managed by the data center operator, rather than by an outside party. While overlays can run across the Internet as well, they will extend well into the data center itself (e.g., up to and including hypervisors) and include large numbers of machines within the data center itself.

Other L2VPN approaches, such as L2TP [RFC2661] require significant tunnel state at the encapsulating and decapsulating end points. Overlays require less tunnel state than other approaches, which is important to allow overlays to scale to hundreds of thousands of end points. It is assumed that smaller switches (i.e., virtual switches in hypervisors or the adjacent devices to which VMs connect) will be part of the overlay network and be responsible for encapsulating and decapsulating packets.

#### 4.7. Proxy Mobile IP

Proxy Mobile IP [RFC5213] [RFC5844] makes use of the GRE Key Field [RFC5845] [RFC6245], but not in a way that supports multi-tenancy.

#### 4.8. LISP

LISP[I-D.ietf-lisp] essentially provides an IP over IP overlay where the internal addresses are end station Identifiers and the outer IP addresses represent the location of the end station within the core IP network topology. The LISP overlay header uses a 24-bit Instance ID used to support overlapping inner IP addresses.

### 5. Further Work

It is believed that overlay-based approaches may be able to reduce

the overall amount of flooding and other multicast and broadcast related traffic (e.g, ARP and ND) currently experienced within current data centers with a large flat L2 network. Further analysis is needed to characterize expected improvements.

There are a number of VPN approaches that provide some if not all of the desired semantics of virtual networks. A gap analysis will be needed to assess how well existing approaches satisfy the requirements.

## 6. Summary

This document has argued that network virtualization using overlays addresses a number of issues being faced as data centers scale in size. In addition, careful study of current data center problems is needed for development of proper requirements and standard solutions.

Three potential work were identified. The first involves the interaction that take place when a VM attaches or detaches from an overlay. A second involves the protocol an NVE would use to communicate with a backend "oracle" to learn and disseminate mapping information about the VMs the NVE communicates with. The third potential work area involves the backend oracle itself, i.e., how it provides failover and how it interacts with oracles in other domains.

## 7. Acknowledgments

Helpful comments and improvements to this document have come from John Drake, Ariel Hendel, Vinit Jain, Thomas Morin, Benson Schliesser and many others on the mailing list.

## 8. IANA Considerations

This memo includes no request to IANA.

## 9. Security Considerations

TBD

## 10. Informative References

[I-D.fang-vpn4dc-problem-statement]  
Napierala, M., Fang, L., and D. Cai, "IP-VPN Data Center

Problem Statement and Requirements",  
draft-fang-vpn4dc-problem-statement-01 (work in progress),  
June 2012.

[I-D.ietf-l2vpn-evpn]

Sajassi, A., Aggarwal, R., Henderickx, W., Balus, F.,  
Isaac, A., and J. Uttaro, "BGP MPLS Based Ethernet VPN",  
draft-ietf-l2vpn-evpn-01 (work in progress), July 2012.

[I-D.ietf-lisp]

Farinacci, D., Fuller, V., Meyer, D., and D. Lewis,  
"Locator/ID Separation Protocol (LISP)",  
draft-ietf-lisp-23 (work in progress), May 2012.

[I-D.ietf-trill-fine-labeling]

Eastlake, D., Zhang, M., Agarwal, P., Perlman, R., and D.  
Dutt, "TRILL: Fine-Grained Labeling",  
draft-ietf-trill-fine-labeling-01 (work in progress),  
June 2012.

[I-D.kreeger-nvo3-overlay-cp]

Kreeger, L., Dutt, D., Narten, T., Black, D., and M.  
Sridhavan, "Network Virtualization Overlay Control  
Protocol Requirements", draft-kreeger-nvo3-overlay-cp-01  
(work in progress), July 2012.

[I-D.lasserre-nvo3-framework]

Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y.  
Rekhter, "Framework for DC Network Virtualization",  
draft-lasserre-nvo3-framework-03 (work in progress),  
July 2012.

[I-D.raggarwa-data-center-mobility]

Aggarwal, R., Rekhter, Y., Henderickx, W., Shekhar, R.,  
and L. Fang, "Data Center Mobility based on BGP/MPLS, IP  
Routing and NHRP", draft-raggarwa-data-center-mobility-03  
(work in progress), June 2012.

[RFC2661]

Townesley, W., Valencia, A., Rubens, A., Pall, G., Zorn,  
G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"",  
RFC 2661, August 1999.

[RFC4364]

Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private  
Networks (VPNs)", RFC 4364, February 2006.

[RFC5213]

Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K.,  
and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.

- [RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", RFC 5844, May 2010.
- [RFC5845] Muhanna, A., Khalil, M., Gundavelli, S., and K. Leung, "Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6", RFC 5845, June 2010.
- [RFC6245] Yegani, P., Leung, K., Lior, A., Chowdhury, K., and J. Navali, "Generic Routing Encapsulation (GRE) Key Extension for Mobile IPv4", RFC 6245, May 2011.
- [RFC6325] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.
- [SPBM] "IEEE P802.1aq/D4.5 Draft Standard for Local and Metropolitan Area Networks -- Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks, Amendment 8: Shortest Path Bridging", February 2012.

## Appendix A. Change Log

### A.1. Changes from -01

1. Removed Section 4.2 (Standardization Issues) and Section 5 (Control Plane) as those are more appropriately covered in and overlap with material in [I-D.lasserre-nvo3-framework] and [I-D.kreeger-nvo3-overlay-cp].
2. Expanded introduction and better explained terms such as tenant and virtual network instance. These had been covered in a section that has since been removed.
3. Added Section 3.3 "Overlay Networking Work Areas" to better articulate the three separable work components (or "on-the-wire protocols") where work is needed.
4. Added section on Shortest Path Bridging in Related Work section.
5. Revised some of the terminology to be consistent with [I-D.lasserre-nvo3-framework] and [I-D.kreeger-nvo3-overlay-cp].

### A.2. Changes from -02

1. Numerous changes in response to discussions on the nvo3 mailing list, with majority of changes in Section 2 (Problem Details) and Section 3 (Network Overlays). Best to see diffs for specific

text changes.

#### A.3. Changes from -03

1. Too numerous to enumerate; moved solution-specific descriptions to Related Work section. Pulled in additional text (and authors) from from [I-D.fang-vpn4dc-problem-statement], numerous editorial improvements.

#### Authors' Addresses

Thomas Narten (editor)  
IBM

Email: narten@us.ibm.com

David Black  
EMC

Email: david.black@emc.com

Dinesh Dutt

Email: ddutt.ietf@hobbesdutt.com

Luyuan Fang  
Cisco Systems  
111 Wood Avenue South  
Iselin, NJ 08830  
USA

Email: lufang@cisco.com

Eric Gray  
Ericsson

Email: eric.gray@ericsson.com

Lawrence Kreeger  
Cisco

Email: kreeger@cisco.com

Maria Napierala  
AT&T  
200 Laurel Avenue  
Middletown, NJ 07748  
USA

Email: mnapierala@att.com

Murari Sridharan  
Microsoft

Email: muraris@microsoft.com





Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 05, 2014

YJ. Stein  
Y. Gittik  
RAD Data Communications  
D. Kofman  
K. Katsaros  
LINCS  
M. Morrow  
L. Fang  
Cisco Systems  
W. Henderickx  
Alcatel-Lucent  
July 04, 2013

Accessing Cloud Services  
draft-stein-cloud-access-03.txt

Abstract

Cloud services are revolutionizing the way computational resources are provided, but at the expense of requiring an even more revolutionary overhaul of the networking infrastructure needed to deliver them. Much recent work has focused on intra- and inter-datacenter connectivity requirements and architectures, while the "access segment" connecting the cloud services user to the datacenter still needs to be addressed. In this draft we consider tighter integration between the network and the datacenter, in order to improve end-to-end Quality of Experience, while minimizing both networking and computational resource costs.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 05, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Model of Existing Cloud Services . . . . .	4
3. Optimized Cloud Access . . . . .	6
4. Security Considerations . . . . .	8
5. IANA Considerations . . . . .	9
6. Acknowledgements . . . . .	9
7. References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

Cloud services replace computational power and storage resources traditionally located under the user's table or on the user's in-house servers, with resources located in remote datacenters. The cloud resources may be raw computing power and storage (Infrastructure as a Service - IaaS), or computer systems along with supported operating systems and tools (Platform as a Service - PaaS), or even fully developed applications (Software as a Service - SaaS). Processing power required for the operation of network devices can also be provided (e.g., Routing as a Service - RaaS). The inter- and intra-datacenter networking architectures needed to support cloud services are described in [I-D.bitar-datacenter-vpn-applicability].

The advantages of cloud services over conventional IT services include elasticity (the ability to increase or decrease resources on demand rather than having to purchase enough resources for worst case scenarios), scalability (allocating multiple resources and load-balancing them), high-availability (resources may be backed up by similar resources at other datacenters), and offloading of IT tasks (such as applications upgrading, firewalling, load balancing, storage backup, and disaster recovery). These translate to economic

efficiencies if actually delivered. The disadvantages of cloud service are lack of direct control by the customer, insecurity regarding remote storage of sensitive data, and communications costs (both direct monetary and technical such as lack of availability and additional transaction latency).

The cloud service user connects to cloud resources over a networking infrastructure. Today this infrastructure is often the public Internet, but (for reasons to be explained below) is preferably a network maintained by a Network Service Provider (NSP). The datacenter(s) may belong to the NSP (which is the case considered by [I-D.masum-chari-shc]), or may belong to a separate Cloud Service Provider (CSP), and accessible from the NSP's network. In the latter case there may or not be a business relationship between the NSP and CSP, the strongest such relationship being when either the NSP or CSP offers a unified "bundled" service to the customer.

In order to obtain the advantages of cloud service without many of the disadvantages, the cloud services customer enters into a Service Level Agreement (SLA) with the CSP. However, such an SLA by itself will be unable to guarantee end-to-end service goals, since it does not cover degradations introduced by the intervening network. Indeed, if the datacenter is accessed over the public Internet, end-to-end service goals may be unattainable. Thus an additional SLA with the NSP (that may already be in effect for pre-cloud services) is typically required. When the CSP and the NSP are the same entity but not offering a bundled service, these SLAs may still be separate documents.

Cloud services require a fundamental rethinking of the Information Technology (IT) infrastructure, due to the requirement for dynamic changes in IT resource configuration. Physical IT resources are replaced by virtualized ones packaged in Virtual Machines (VMs). VMs can be created, relocated while running (VM migration), and destroyed on-demand. Since VMs need to interconnect, connect to physical resources, and connect to the cloud services user, they need to be allocated appropriate IP and layer 2 addresses. Since these addresses need to be allocated, moved, and destroyed on-the-fly, the cloud IT revolution directly impacts the networking infrastructure. Recent work, such as [I-D.bitar-datacenter-vpn-applicability], has focused on requirements and architectures for connectivity inside and between datacenters. However, the "access segment", that is, the networking infrastructure connecting the cloud services user to the datacenter, has not been fully addressed.

The allocation, management, manipulation, and release of cloud resources is called "orchestration" (see [I-D.dalela-orchestration]). Orchestrators need to respond to user demands and uphold user SLAs

(perhaps exploiting virtualization techniques such as VM migration) while taking into account the location and availability of IT resources, and optimizing the CSP's operational objectives. These objectives include, for example, decreasing costs by consolidating resources, balancing use of resources by reallocating computational and storage resources, and enforcing engineering, business, and security policies. Orchestrators of the present generation do not attempt optimization of CSP's networking resources, but this generalization is being studied [I-D.ietf-nvo3-framework]. Furthermore, these orchestrators are completely oblivious to the NSP's resources and objectives. Hence, there is no mechanism for maintaining end-to-end SLAs, or for optimizing end-to-end networking.

This goal of this Internet Draft is to kick off discussions on requirements and possible mechanisms for improving end-to-end Quality of Experience while minimizing both networking and computational costs.

## 2. Model of Existing Cloud Services

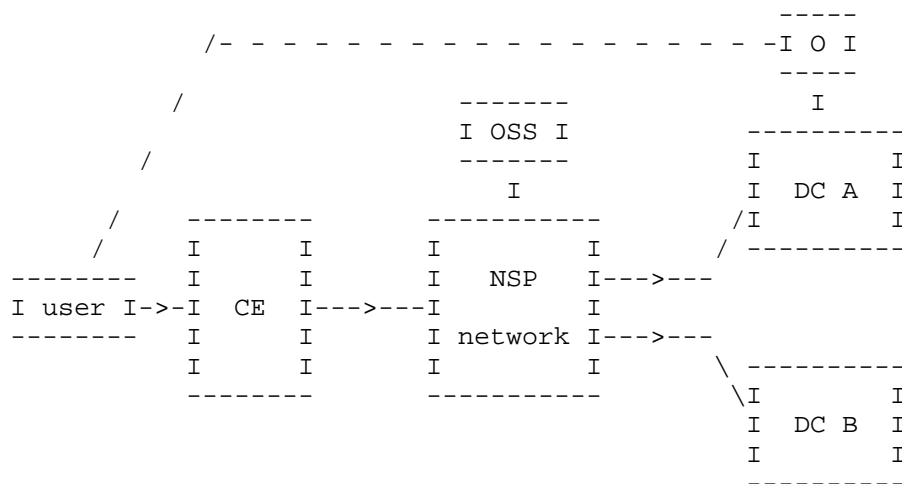


Figure 1: Simplified model of cloud service provided over Service Provider network to an enterprise customer behind a CE device

For concreteness, we will assume the scenario of Figure 1. On the left we see a cloud services user attached to a customer site network. This network connects to the outside world via a Customer Edge (CE), which may be a branch-site router or switch, a special purpose cloud demarcation device, or in degenerate cases the user's computer itself. The NSP network is assumed to be a well-engineered

network providing VPN and other SLA-based services to the customer site. The NSP network is managed from an Operations Support System (OSS), which may include a Business Support System (BSS), the latter being needed for interfacing with the customer for approval of service reconfiguration, billing issues, etc. In some cases, the functionality needed here may be obtained by interfacing with a Looking Glass server or a Policy and Charging Rules Function (PCRF). Connected to this network are datacenters (two are shown - datacenter A and datacenter B), which may belong to the NSP, or to a separate CSP. The orchestrator of datacenter A is depicted as "O". Additionally, Internet access may be available directly from the CE (not shown) or from the NSP network.

In the usual cloud services orchestration model the user requests a well-defined resource, for example over the telephone, via a web-based portal, or via a function call. The orchestrator, after checking correctness, availability, and updating the billing system, allocates the resource, e.g., a VM on a particular CPU located in a particular rack in datacenter A. In addition, the required networking resources are allocated to the VM, e.g., an IP address, an Ethernet MAC address, and a VLAN tag. The VM is now started and consumes CPU power, memory, and disk space, as well as communications bandwidth between itself and other VMs on the same CPU, within the same rack, on other racks in the same datacenter, between datacenters, and between itself and the user. If it becomes necessary to move the VM from its allocated position to somewhere else (VM migration), the orchestrator needs to reallocate the required computational and communications resources. An example case is "cloudbursting" where a customer who finds himself temporarily with insufficient local resources reaches out to the cloud for supplementary ones [I-D.mcdysan-sdnp-cloudbursting-usecase]. A priori this requires allocating new addresses and rerouting all of the aforementioned traffic types, while maintaining continuous operation of the VM. When the user informs the CSP that it no longer requires the VM, the orchestrator needs to clear the routing entries, withdraw the communications resources, release storage and computational resources, and update the billing system.

The operations of the previous paragraph are all performed by the orchestrator, with possible cooperation with orchestrators from other datacenters. The needed routing information is advertised to the NSP via standard routing protocols, without taking into account possible effects on the NSP network. If, for example, the path in the NSP network to datacenter A degrades, while the path to datacenter B is performing well, this information is neither known by the orchestrator, nor is there a method for the orchestrator to take it into account. Instead, the NSP must find a way to reach datacenter A, even if this path is expensive, or of high latency, or problematic in some other way.

This predicament arises due to the orchestrator communicating (indirectly) with the user, but not with the NSP's OSS. In addition, although the CE may be capable of OAM functionality, fault and performance monitoring of the communications path through the NSP network are not employed. Finally, while the user can (indirectly) communicate with the orchestrator, there is no coordinated path to the NSP's OSS/BSS.

### 3. Optimized Cloud Access

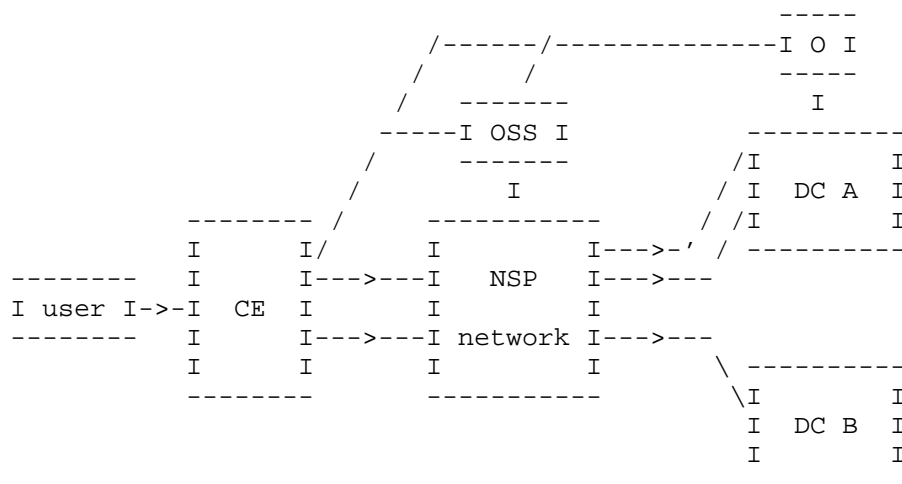


Figure 2: Cloud service with dual homing between a cloud-aware CE and NSP network, and coordination between CE, NSP OSS/BSS, and orchestrator

Figure 2. depicts two enhancements to the previous scenario. The trivial enhancement is the providing of dual-homing between the CE and the NSP network. This is a well-known and widely deployed

feature, which may be implemented regardless of the cloud services. We shall see that it acquires additional meaning in the context of the solution described below.

More significantly, Figure 2 depicts three new control communications channels. The CE device is now assumed to be cloud-aware, and may communicate directly with the NSP OSS/BSS, and with the CSP orchestrator. In addition, the latter two may communicate with each other. These control channels facilitate new capabilities, that may improve end-to-end QoE while optimizing operational cost. An alternative to a combined cloud/network CE is a separate "cloud demarcation device" placed behind the network CE.

Consider the provisioning of a new cloud service. With this new architecture the user's request is proxied by the cloud-aware CE to both the OSS/BSS and to the orchestrator. Before commissioning the service, the orchestrator initiates network testing between the datacenter and the CE, and with the NSP's assistance QoS parameters are determined for alternative paths to various relevant datacenters. The NSP and CSP (whether a single SP or two) can now jointly decide on placement of the VM in order to optimize the user's end-to-end Quality of Experience (QoE) while minimizing costs to both SPs. The best placement will necessitate the solution of a joint CSP + NSP optimization problem, while the latter minimization may only be reliable when a single SP provides networking and cloud resources. The joint optimization calculation will input the status of computational and storage resources at all relevant datacenters; as well as network delay, throughput, and packet loss to each datacenter. In some cases re-allocation of existing computational and networking resources may be needed.

Similarly, the NSP OSS may trigger VM migration if network conditions degrade to the point where user QoE is no longer at the desired level, or may veto a CSP initiated VM migration when its effect would be too onerous on the NSP network.

The cloud-aware CE may be configured to periodically test path continuity and measure QoS parameters. The CE can then report that the estimated QoE drops under that specified in the SLA (or dangerously approaches it), in order to promote SLA assurance even when neither OSS nor orchestrator would otherwise know of the problem. Additionally, the cloud-aware CE may report workload changes detected by monitoring the number of active sessions (e.g., the number of "flows" or n-tuple pairs). The OSS and orchestrator can jointly perform root cause analysis and decide to trigger VM migration or network allocation changes or both. Finally, over-extended network segments may be identified, and pro-active VM migration and/or rerouting performed to better distribute the load.



When the CE is dual homed to the NSP network, the secondary link may be utilized in the conventional manner when the primary link fails, or may be selected as part of the overall optimization of QoE vs. cost. Load balancing over both links may also be employed. The datacenters may also be connected to the network with multiple links (as depicted for DC A in Figure 2), enabling further connectivity optimization.

In addition, popular yet stationary content may be cached in the NSP network, and optimization may lead to the NSP network providing this content without the need to access the datacenter at all. In certain cases (e.g., catastrophic failure in the NSP network or of the connectivity between that network and the datacenter), the cloud-aware CE may choose to bypass the NSP network altogether and reach the datacenter over the public Internet (with consequent QoE reduction). In other cases, it may make sense to locally provide standalone resources at the cloud demarcation device itself.

#### 4. Security Considerations

Perceived insecurity of the customer's data sent to the cloud or stored in a datacenter is perhaps the single most important factor impeding the wide adoption of cloud services. At present, the only solutions have been end-to-end authentication and confidentiality, with the high cost these place on user equipment. The cloud-aware CE may assume the responsibility for securing the cloud services from the edge of the customer's walled garden, all the way to the datacenter.

Isolation of CSP customers is addressed in [I-D.masum-chari-shc]. Security measures such as hiding of network topology, as well as on-the-fly inspection and modification of transactions are listed as requirements in [I-D.dalela-orchestration], while [I-D.dalela-sop] specifies encryption and authentication of orchestration protocol messages.

A further extension to the model is to explicitly include security levels as parameters of the QoE optimization process. This parameter may be relatively coarse-grained (for example, 1 for services which must be provided only over secure links, 0.5 for those for which access paths under direct control of the NSP is sufficient, 0 for general services that may run over out-of-footprint connections). Security may also take regulatory restrictions into account, such as limitations on database migration across national boundaries. Thus, the placement and movement of a VM will be accomplished based on full optimization of computational and storage resources; network delay, throughput, and packet loss; and security levels. For example, for an application for which the user can not afford denial of service

the joint optimization would need to find the needed resources as close as possible to the end user.

## 5. IANA Considerations

This document requires no IANA actions.

## 6. Acknowledgements

The work of Y(J)S, YG, DK, and KK was conducted under the aegis of ETICS (Economics and Technologies for Inter-Carrier Services), a European collaborative research project within the ICT theme of the 7th Framework Programme of the European Union that contributes to the objective "Network of the Future".

## 7. References

- [I-D.bitar-datacenter-vpn-applicability]  
Bitar, N., Balus, F., Lasserre, M., Henderickx, W., Sajassi, A., Fang, L., Ikejiri, Y., and M. Pisica, "Cloud Networking: Framework and VPN Applicability", draft-bitar-datacenter-vpn-applicability-02 (work in progress), May 2012.
- [I-D.bitar-datacenter-vpn-applicability]  
Bitar, N., Balus, F., Lasserre, M., Henderickx, W., Sajassi, A., Fang, L., Ikejiri, Y., and M. Pisica, "Cloud Networking: Framework and VPN Applicability", draft-bitar-datacenter-vpn-applicability-02 (work in progress), May 2012.
- [I-D.dalela-orchestration]  
Dalela, A. and M. Hammer, "Service Orchestration Protocol (SOP) Requirements", draft-dalela-orchestration-00 (work in progress), January 2012.
- [I-D.dalela-sop]  
Dalela, A. and M. Hammer, "Service Orchestration Protocol", draft-dalela-sop-00 (work in progress), January 2012.
- [I-D.masum-chari-shc]  
Hasan, M., Chari, A., Fahed, D., Tucker, L., Morrow, M., and M. Malyon, "A framework for controlling Multitenant Isolation, Connectivity and Reachability in a Hybrid Cloud Environment", draft-masum-chari-shc-00 (work in progress), February 2012.
- [I-D.mcdysan-sdn-cloudbursting-usecase]  
McDysan, D., "Cloud Bursting Use Case", draft-mcdysan-sdn-cloudbursting-usecase-00 (work in progress), October 2011.
- [I-D.ietf-nvo3-framework]  
Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-02 (work in progress), February 2013.

Authors' Addresses

Yaakov (Jonathan) Stein  
RAD Data Communications  
24 Raoul Wallenberg St., Bldg C  
Tel Aviv 69719  
Israel

Email: [yaakov\\_s@rad.com](mailto:yaakov_s@rad.com)

Yuri Gittik  
RAD Data Communications  
24 Raoul Wallenberg St., Bldg C  
Tel Aviv 69719  
Israel

Email: [yuri\\_g@rad.com](mailto:yuri_g@rad.com)

Daniel Kofman  
LINCS  
23 Avenue d'Italie  
Paris 75013  
France

Email: [daniel.kofman@telecom-paristech.fr](mailto:daniel.kofman@telecom-paristech.fr)

Konstantinos Katsaros  
LINCS  
23 Avenue d'Italie  
Paris 75013  
France

Email: [katsaros@telecom-paristech.fr](mailto:katsaros@telecom-paristech.fr)

Monique Morrow  
Cisco Systems  
Richtistrasse 7  
CH-8304 Wallisellen  
Switzerland

Email: [mmorrow@cisco.com](mailto:mmorrow@cisco.com)

Luyuan Fang  
Cisco Systems  
300 Beaver Brook Road  
Boxborough, MA 01719  
US

Email: [lufang@cisco.com](mailto:lufang@cisco.com)

Wim Henderickx  
Alcatel-Lucent  
Copernicuslaan 50  
2018 Antwerp  
Belgium

Email: [wim.henderickx@alcatel-lucent.com](mailto:wim.henderickx@alcatel-lucent.com)

NV03 Working Group  
Internet Draft  
Intended status: Informational  
Expires: January 16, 2013

Y. Wei, Ed.  
ZTE Corporation  
L. Fang, Ed.  
Cisco Systems  
S. Zhang  
ZTE Corporation

July 16, 2012

NV03 Security Framework  
draft-wei-nv03-security-framework-01

Abstract

This document provides a security framework for overlay based network virtualization. It describes the security reference model, the security threats and security requirements.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

This Internet-Draft will expire on January 16, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction .....	2
2. Terminologies .....	3
3. Security Reference Models .....	5
3.1. Scenario 1: Virtual Network and DC infrastructure belong to the same DC operator .....	5
4. Security Threats .....	6
4.1. Attacks on Control Plane .....	7
4.2. Attacks on the Data Plane .....	7
5. Security Requirements .....	8
5.1. Control Plane Security Requirements .....	8
5.2. Data Plane Security Requirements .....	8
6. Security Considerations .....	8
7. IANA Considerations .....	9
8. Normative References .....	9
9. Informative References .....	9
10. Author's Addresses .....	10

## Requirements Language

Although this document is not a protocol specification, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC

2119].

## 1. Introduction

[Page 2]



Security is one of most important factors in the environment of cloud computing and particularly to the Data Center Network Virtualization where multi-tenancy support is one of the fundamental requirements.

Though security considerations are provided in several individual documents, a general description of security for NVO3 is needed, especially there are areas not covered in the current documents. The motivation of this document is to provide a general and consistent security framework for NVO3, and to provide a guidance for the design of related protocol.

This document is organized as follows. Section 3 describes the security reference model in the context of NVO3, which defines which component is trusted or not for a particular scenario. Section 4 describes the security threats under the security model. Section 5 addresses the security requirements in response to the security threats.

## 2. Terminologies

This document uses NVO3 specific terminology defined in [I-D.lasserre-nvo3-framework]. For reader's convenience, this document repeats some of the definitions in addition to reference it.

**NVE:** Network Virtualization Edge. It is a network entity that sits on the edge of the NVO3 network. It implements network virtualization functions that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses). An NVE could be implemented as part of a virtual switch within a hypervisor, a physical switch or router, a Network Service Appliance or even be embedded within an End Station.

**VN:** Virtual Network. This is a virtual L2 or L3 domain that belongs a tenant.

**VNI:** Virtual Network Instance. This is one instance of a virtual overlay network. Two Virtual Networks are isolated from one another and may use overlapping addresses.

**Virtual Network Context or VN Context:** Field that is part of the overlay encapsulation header which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE. The egress NVE uses this field to determine the appropriate virtual network context in which to process the packet. This field MAY be an explicit, unique (to the administrative domain) virtual network identifier (VNID) or MAY express the

necessary context information in other ways (e.g. a locally significant identifier).

**VNID:** Virtual Network Identifier. In the case where the VN context has global significance, this is the ID value that is carried in each data packet in the overlay encapsulation that identifies the Virtual Network the packet belongs to.

**Underlay or Underlying Network:** This is the network that provides the connectivity between NVEs. The Underlying Network can be completely unaware of the overlay packets. Addresses within the Underlying Network are also referred to as "outer addresses" because they exist in the outer encapsulation. The Underlying Network can use a completely different protocol (and address family) from that of the overlay.

**Data Center (DC):** A physical complex housing physical servers, network switches and routers, Network Service Appliances and networked storage. The purpose of a Data Center is to provide application and/or compute and/or storage services. One such service is virtualized data center services, also known as Infrastructure as a Service.

**Virtual Data Center or Virtual DC:** A container for virtualized compute, storage and network services. Managed by a single tenant, a Virtual DC can contain multiple VNs and multiple Tenant End Systems that are connected to one or more of these VNs.

**VM:** Virtual Machine. Several Virtual Machines can share the resources of a single physical computer server using the services of a Hypervisor (see below definition).

**Hypervisor:** Server virtualization software running on a physical compute server that hosts Virtual Machines. The hypervisor provides shared compute/memory/storage and network connectivity to the VMs that it hosts. Hypervisors often embed a Virtual Switch (see below).

**Virtual Switch:** A function within a Hypervisor (typically implemented in software) that provides similar services to a physical Ethernet switch. It switches Ethernet frames between VMs' virtual NICs within the same physical server, or between a VM and a physical NIC card connecting the server to a physical Ethernet switch. It also enforces network isolation between VMs that should not communicate with each other.

**Tenant:** A customer who consumes virtualized data center services offered by a cloud service provider. A single tenant may consume

one or more Virtual Data Centers hosted by the same cloud service provider.

Tenant End System: It defines an end system of a particular tenant, which can be for instance a virtual machine (VM), a non-virtualized server, or a physical appliance.

### 3. Security Reference Models

This section defines the security reference models for Overlay based Network Virtualization.

The L3 overlay network provides virtual network to multi-tenants, which is deployed on the underlying network. The tenant end system attaches to the L3 overlay network.

L3 overlay network provides isolation to each tenant, which provides a level of security to its tenant. L3 overlay network can be regarded secure zone from the view of ONV3 operator. Other components outside of the ONV3 are considered as untrusted, which may impose security risk to the NVO3 networks. Each virtual network should assume not trusting other virtual networks. This model is the basis to analyze the security of ONV3.

3.1. Scenario 1: Virtual Network and DC infrastructure belong to the same DC operator

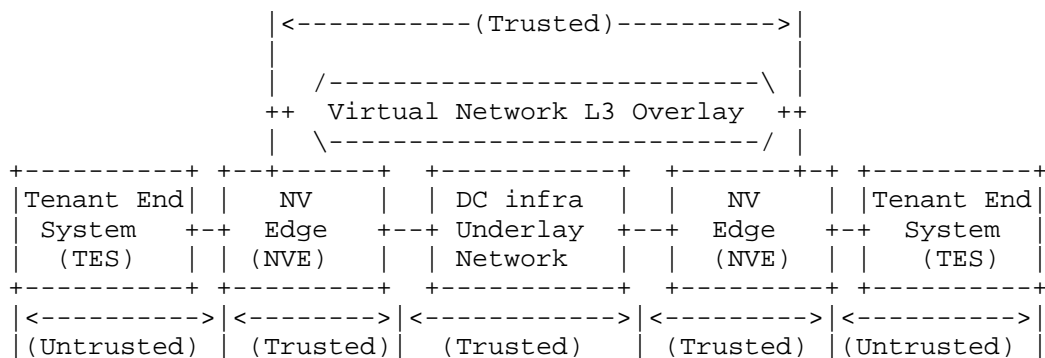


Figure 1. Trust Model for Scenario 1

Notes:

- 1) The diagram is a logical illustration. The TES and NVE may be implemented in the same physical device, or separate devices.
- 2) The physical end system may in reality include virtual switching/routing instances and multiple VMS (TESs) belong to different tenants.
- 3) The trusted or untrusted notions in the diagram is from a Virtual Overlay Network point of view, not the underlay network.
- 4) Each VN treats other VNs as untrusted.

Scenario 2: Virtual Network and DC infrastructure belong to different DC operators

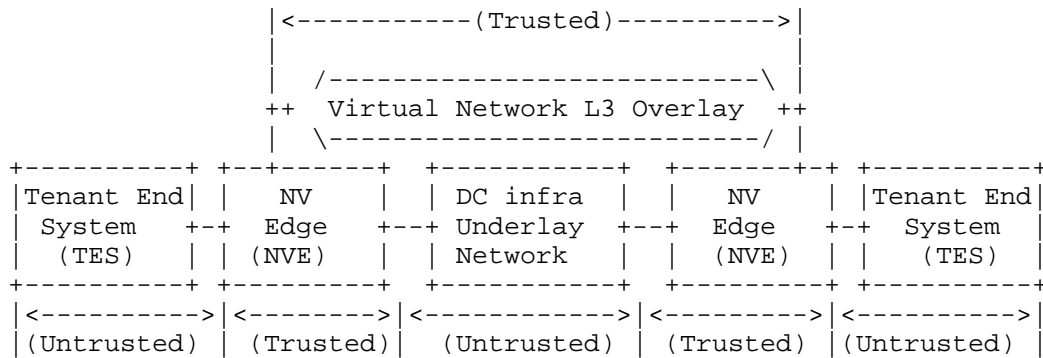


Figure 2. Trust Model for Scenario 2

The same notes listed under scenario 1 also apply here.

#### 4. Security Threats

This section describes the various security threats that may endanger overlay based network virtualization. For example, an attack on ONV3 may result in some unexpected effects:

- o Interrupt the connectivity of tenant's virtual network.
- o Inject some unwanted traffic into virtual network.
- o Eavesdrop sensitive information from tenant.
- o Degrade provider's service level.

Security threats may be malicious or casual. For example, some of them may come from the following sources:

- o A tenant who rents one or more virtual networks may want to acquire some information from other tenants co-existed in the same data center.
- o Some persons who manipulate the activation, migration or deactivation of tenant's virtual machine.
- o Some persons who physically access to underlying network.

#### 4.1. Attacks on Control Plane

1. Attack association between VM and VN: one of the functionalities of ONV3 is to provide virtual network to multi-tenants. ONV3 associates a virtual machine's NIC with corresponding virtual network, and maintain that association as the VM is activated, migrated or deactivated. The signaling information between endpoint and access switch may be spoofed or altered. Thus the association between VM and VN may be invalid if the signaling is not properly protected.

2. Attack the mapping of a virtual network: The mapping between the inner and outer addresses may be affected through altering the mapping table.

3. Inject traffic: The comprised underlying network may inject traffic into virtual network.

4. Attack live migration: An attacker may cause guest VMs to be live migrated to the attacker's machine and gain full control over guest VMs.

5. Denial of Service attacks against endpoint by false resource advertising: for live migration initiated automatically to distribute load across a number of servers, an attacker may falsely advertise available resources via the control plane. By pretending to have a large number of spare CPU cycles, that attacker may be able to influence the control plane to migrate a VM to a compromised endpoint.

#### 4.2. Attacks on the Data Plane

1. Unauthorized snooping of data traffic: This is attack results in leakage of sensitive information, attacker can sniff information from the user packets and extract their content.

2. Modification of data traffic: An attacker may modify, insert or delete data packets and impersonate them as legitimate ones.

3. Man-in-the-Middle attack on live migration of VM: When a virtual machine is migrated from one endpoint to another, the VM may be intercepted and modified in the middle of the migration.

## 5. Security Requirements

This section describes security requirements for control plane and data plane of NVO3.

### 5.1. Control Plane Security Requirements

1. The network infrastructure shall support mechanisms for authentication and integrity protection of the control plane. (1)When a protocol is used for the service auto-provisioning/discovery, the information from endpoint shall not be spoofed or altered. (2)When a protocol is used to distribute address advertisement and tunneling information, the protocol shall provide integrity protection. (3)The protocol for tunnel management shall provide integrity and authentication protection.
2. NVEs shall assure the information in the mapping table is coming from a trusted source.
3. The virtual network should prevent malformed traffic injection from underlying network, other virtual network, or endpoint.

### 5.2. Data Plane Security Requirements

1. The mapping function from the tenant to overlay shall be protected. NVEs should verify VNID is not spoofed.
2. The data plane should protect VM's state against snooping and tampering.
3. IPsec can be used to provide authentication, integrity and confidentiality protection. IPsec can be used to protect the data plane.

## 6. Security Considerations

NVO3 Security Framework  
Expires Jan. 2013

This document discusses general security threats and requirements for NVO3. Individual document may raise specific issues based on the particular content and should address them in the individual document.

## 7. IANA Considerations

This document contains no new IANA considerations.

## 8. Normative References

## 9. Informative References

[Editors' note: All Network Virtualization with Layer 3 (NVO3) Internet drafts or related ID(s) referenced in the following list are currently work in progress individual drafts in their early development stage, status and text are subject to change, more reference may be added along with the development of the NVO3 WG.]

[RFC 2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4111] Fang, L., Fang, L., Ed., "Security Framework for Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4111, July 2005.

[RFC5920] Fang, L., "Security Framework for MPLS and GMPLS Networks", RFC 5920, July 2010.

[opsec-efforts] Lonvick, C., and Spak, D., "Security Best Practices Efforts and Documents", IETF draft-ietf-opsec-efforts-18.txt, April 2012.

[VM-Migration] Oberheide, Jon., Cooke, Evan., and Farnam. Jahanian, "Empirical Exploitation of Live Virtual Machine Migration", Feb 2011.

[I-D.narten-nvo3-overlay-problem-statement] Narten, T., Sridhavan, M., Dutt, D., Black, D., and L. Kreeger, "Problem Statement: Overlays for Network Virtualization", draft-narten-nvo3-overlay-problem-statement-02 (work in progress), June 2012.

[I-D.fang-vpn4dc-problem-statement] Napierala M., Fang L., Cai, Dennis, "IP-VPN Data Center Problem Statement and Requirements", draft-fang-vpn4dc-problem-statement-01.txt, June 2012.

NVO3 Security Framework  
Expires Jan. 2013

[I-D.lasserre-nvo3-framework] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y.Rekhter, "Framework for DC Network Virtualization", draft-lasserre-nvo3-framework-03 (work in progress), July 2012.

[I-D.kreeger-nvo3-overlay-cp] Black, D., Dutt, D., Kreeger, L., Sridhavan, M., and T. Narten, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp-00 (work in progress), January 2012.

[I-D.bitar-lasserre-nvo3-dp-reqs] Bitar, N., Lasserre, M., and F. Balus, "NVO3 Data Plane Requirements", draft-bitar-lasserre-nvo3-dp-reqs-00 (work in progress), May 2012.

#### 10. Author's Addresses

Yinxing Wei (Editor)  
ZTE Corporation  
No 68, Zijinghua Road  
Nanjing, Jiangsu 210012  
China

Phone: +86 25 52872328  
Email: wei.yinxing@zte.com.cn

Luyuan Fang (Editor)  
Cisco Systems, Inc.  
111 Wood Ave. South  
Iselin, NJ 08830  
USA  
Email: lufang@cisco.com

Shiwei Zhang  
ZTE Corporation  
No 68, Zijinghua Road  
Nanjing, Jiangsu 210012  
China

Phone: +86 25 52870100  
Email: zhang.shiwei@zte.com.cn